aws

버전 1 사용 설명서

# AWS Command Line Interface

# AWS Command Line Interface: 버전 1 사용 설명서

# Table of Contents

이 설명서는 의 버전 1 AWS CLI 전용입니다. 의 버전 2와 관련된 설명서는 버전 2 사용 설명서 를 AWS CLI참조하세요.

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.

# AWS Command Line Interface 버전 1이란 무엇인가요?

> **Note**
>
> AWS CLI 버전 1은 의 최신 버전이 아닙니다 AWS CLI. AWS CLI 버전 2에 도입된 일부 기능은 버전 1과 백포트되지 않으므로 이러한 기능에 액세스하려면 업그레이드해야 합니다. 버전 1과 "호환되지 않는" 일부 변경 사항이 있으므로 스크립트를 변경해야 할 수 있습니다. 버전 2에 도입된 호환성에 영향을 미치는 변경 사항 목록은 AWS CLI 버전 2 사용 설명서에서 호환성에 영향을 미치는 변경을 참조하세요.

AWS Command Line Interface (AWS CLI) 는 명령줄 셸의 명령을 사용하여 AWS 서비스와 상호 작용할 수 있는 오픈 소스 도구입니다. 최소 구성으로 AWS CLI 를 사용하면 터미널 프로그램의 명령 프롬프트에서 브라우저 AWS Management Console 기반에서 제공하는 것과 동일한 기능을 구현하는 명령을 실행할 수 있습니다.

- Linux 셸 - `bash`, `zsh`, `tcsh` 등의 일반적인 셸 프로그램을 사용하여 Linux 또는 macOS에서 명령을 실행합니다.
- Windows 명령줄 — Windows에서는 Windows 명령 프롬프트 또는 내부에서 명령을 실행합니다. PowerShell
- 원격 — Pu TTY 또는 SSH with 같은 원격 터미널 프로그램을 통해 Amazon Elastic Compute Cloud (AmazonEC2) 인스턴스에서 명령을 실행합니다 AWS Systems Manager.

의 모든 IaaS (서비스형 인프라) AWS 관리, 관리 및 액세스 기능은 및 에서 사용할 수 있습니다. AWS Management Console AWS API AWS CLI새로운 AWS IaaS 기능 및 서비스는 출시 전, 출시 CLI 시 또는 출시 후 180일 이내에 모든 AWS Management Console 기능을 제공합니다. API

를 AWS CLI 통해 일반 대중에게 직접 액세스할 APIs 수 있습니다 AWS . 를 사용하여 서비스의 기능을 탐색하고 셸 스크립트를 개발하여 리소스를 관리할 수 있습니다. AWS CLI API동일한 수준의 저수준 명령 외에도 여러 AWS 서비스에서 에 대한 사용자 지정 기능을 제공합니다. AWS CLI사용자 지정에는 컴플렉스가 포함된 서비스 사용을 단순화하는 상위 수준 명령이 포함될 수 있습니다. API

## AWS CLI 버전 1 정보

AWS CLI 버전 1은 AWS CLI오리지널이며 계속 지원합니다. 하지만 AWS CLI 버전 2에 도입된 주요 새 기능은 AWS CLI 버전 1로 백포트되지 않을 수 있습니다. 이러한 기능을 사용하려면 AWS CLI 버전

2를 설치해야 합니다. AWS CLI 버전 1은 Python을 SDK 사용하여 빌드되었으므로 호환되는 Python 버전을 설치해야 합니다.

AWS CLI 버전 1을 설치하려면 을 참조하십시오설치 AWS CLI.

다음 명령을 사용하여 현재 설치된 버전을 점검하세요.

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

버전 기록은 의 AWS CLI 버전 1 변경 로그를 참조하십시오. GitHub

## 메이저 버전에 대한 SDK 유지 관리 및 지원

SDK메이저 버전과 기본 종속성에 대한 유지 관리 및 지원에 대한 자세한 내용은 AWS SDKs및 도구 참조 안내서의 다음을 참조하십시오.

- AWS SDKs및 도구 유지 관리 정책
- AWS SDKs및 도구 버전 지원 매트릭스

## Amazon Web Services에 대하여

Amazon Web Services(AWS)는 애플리케이션을 개발할 때 개발자들이 활용할 수 있는 디지털 인프라 서비스의 컬렉션입니다. 서비스에는 컴퓨팅, 스토리지, 데이터베이스 및 애플리케이션 동기화 (메시징 및 대기열) 가 포함됩니다. AWS pay-as-you-go 서비스 모델을 사용합니다. 사용자 또는 애플리케이션이 사용하는 서비스에 대해서만 청구됩니다. 또한 프로토타이핑 및 실험을 위한 플랫폼으로서 접근성을 AWS 높이기 위해 프리 티어를 AWS 제공합니다. 이 계층에서 특정 사용 수준 미만의 서비스는 무료입니다. AWS 비용 및 프리 티어에 대한 자세한 내용은 프리 티어를 참조하십시오.AWS AWS 계정을 만들려면 AWS 홈 페이지를 연 다음 AWS 계정 생성을 선택합니다.

## AWS CLI 사용 설명서의 예제 정보

이 가이드의 AWS Command Line Interface (AWS CLI) 예제는 다음 규칙을 사용하여 형식이 지정됩니다.

- 프롬프트 - 명령 프롬프트는 Linux 프롬프트를 사용하며 ($ )로 표시됩니다. Windows와 관련된 명령의 경우 C:\>가 프롬프트로 사용됩니다. 명령을 입력할 때 프롬프트를 포함시키지 마십시오.

- 디렉터리 - 특정 디렉터리에서 명령을 실행해야 하는 경우 프롬프트 기호 앞에 디렉터리 이름이 표시됩니다.
- 사용자 입력 - 명령줄에 입력하는 명령 텍스트는 **`user input`**으로 형식이 지정됩니다.
- 교체 가능한 텍스트 - 선택한 리소스의 이름을 포함하여 또는 명령에 포함해야 하는 AWS 서비스에서 IDs 생성한 변수 텍스트는 로 형식이 지정됩니다.*`replaceable text`*. 특정 키보드 입력이 필요한 다중 행 명령 또는 명령에서 키보드 명령을 대체 가능한 텍스트로 표시할 수도 있습니다.
- 출력 - AWS 서비스에서 반환한 출력은 사용자 입력 아래에 표시되며 형식이 입니다`computer output`.

다음 **`aws configure`** 명령 예제는 사용자 입력, 대체 가능한 텍스트 및 출력을 보여줍니다.

1. 명령줄에서 **`aws configure`**를 입력한 다음 Enter 키를 누릅니다.
2. 텍스트 줄을 AWS CLI 출력하여 추가 정보를 입력하라는 메시지가 표시됩니다.
3. 각 액세스 키를 차례로 입력한 다음 Enter(입력)를 누릅니다.
4. 그런 다음 표시된 형식으로 AWS 리전 이름을 입력하고 Enter 키를 누른 다음 최종 시간 입력을 눌러 출력 형식 설정을 건너뜁니다.
5. 마지막 Enter(입력) 명령은 해당 줄에 대한 사용자 입력이 없기 때문에 대체 가능한 텍스트로 표시됩니다.

```
$ 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
```

다음의 예는 출력과 간단한 명령을 보여줍니다. 이 예제를 사용하려면 명령의 전체 텍스트를 입력하고 (프롬프트 다음에 강조 표시된 텍스트) Enter(입력)를 누릅니다. 보안 그룹의 이름,*`my-sg`*는 원하는 보안 그룹 이름으로 대체할 수 있습니다. 컬리 브레이스를 포함한 JSON 문서가 출력됩니다. 텍스트 또는 테이블 형식으로 출력CLI하도록 를 구성하면 출력 형식이 다르게 지정됩니다. JSON 는 기본 출력 형식입니다.

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

# 에 대한 추가 설명서 및 리소스 AWS CLI

## AWS CLI 설명서 및 리소스

이 사용 설명서 외에도 를 사용할 때 유용한 온라인 리소스는 다음과 같습니다 AWS CLI.

- AWS CLI 버전 1 참조 가이드
- AWS CLI Bash 스크립팅 코드 예제 리포지토리 . 오픈 소스 bash 스크립팅 예제. Bash 스크립팅 예제는 의 AWS 코드 예제 리포지토리에서 호스팅됩니다GitHub.
- AWS CLI GitHub 리포지토리 . 에서 의 소스 코드를 보고 포크할 수 AWS CLI 있습니다GitHub. 의 사용자 커뮤니티에 가입GitHub하여 피드백을 제공하고, 기능을 요청하고, 자체 기여를 제출하세요. 여기에는 AWS CLI 설명서의 명령 예제 보기 및 제공이 포함됩니다.
- AWS CLI 별칭 예제 리포지토리 에서 및 포크 AWS CLI 별칭 예제를 볼 수 있습니다GitHub.
- AWS CLI 버전 1 Changelog
- AWS CLI 버전 2 Changelog

## 기타 AWS SDKs 및 도구

사용 사례에 따라 필요에 맞게 또는 도구 중 AWS SDKs 하나를 선택할 수 있습니다.

- AWS SDKs 및 도구 참조 가이드
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java
- AWS SDK for JavaScript
- AWS SDK for Kotlin
- AWS SDK for .NET
- AWS SDK for Python (Boto)
- AWS SDK for PHP
- AWS Tools for PowerShell
- AWS SDK for Ruby
- AWS SDK for Rust
- AWS SDK for SAP ABAP

- AWS SDK for Swift
- AWS Amplify

# 설치, 업데이트 및 제거 AWS CLI

이 주제에서는 AWS Command Line Interface (AWS CLI)의 원래 버전을 설치, 업데이트 및 제거하기 위한 링크를 제공합니다. AWS CLI 버전 1은 현재 지원되지만 AWS CLI 버전 2에 추가된 새 기능은 AWS CLI 버전 1에 추가되지 않을 수 있습니다. 이러한 기능을 사용하려면 AWS CLI 버전 2를 설치해야 합니다. 버전 2를 설치하는 방법에 대한 자세한 내용은 AWS CLI 버전 2 설치를 참조하세요.

AWS CLI 설치, 업데이트 및 제거 지침:

- Python 버전 요구 사항
- Amazon Linux에서 AWS CLI 버전 1 설치, 업데이트 및 제거
- Linux에서 AWS CLI 버전 1 설치, 업데이트 및 제거
- macOS에서 AWS CLI 버전 1 설치, 업데이트 및 제거
- Windows에서 AWS CLI 버전 1 설치, 업데이트 및 제거
- 가상 환경에서 AWS CLI 버전 1 설치 및 업데이트

## Python 버전 요구 사항

AWS CLI 버전 1은 SDK for Python을 사용하여 빌드되므로 호환되는 버전의 Python을 설치해야 합니다.

Python 버전 지원 매트릭스

| AWS CLI 버전 | 지원되는 Python 버전 |
|---|---|
| 1.32.0~현재 | Python 3.8 이상 |
| 1.27.0~1.31.x | Python 3.7 이상 |
| 1.20.0~1.26.x | Python 3.6 이상 |
| 1.19.0~1.19.x | Python 2.7 이상, Python 3.6 이상 |
| 1.17 ~ 1.18.x | Python 2.7 이상, Python 3.4 이상 |
| 1.0 ~ 1.16.x | Python 2.6 이상, Python 3.3 이상 |

의 최신 릴리스에 대한 자세한 내용은 의 AWS CLI 버전 2 Changelog를 AWS CLI참조하세요 GitHub.

# Amazon Linux에서 AWS CLI 버전 1 설치, 업데이트 및 제거

AWS CLI 버전 1은 Amazon Linux 및 Amazon Linux 2에 사전 설치되어 있습니다. 다음 명령을 사용하여 현재 설치된 버전을 점검하세요.

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

Amazon Linux 인스턴스를 생성한 시기에 따라 AWS CLI 버전 1은 다음 패키지 관리자 중 하나를 사용하여 사전 설치됩니다.

- pip
- yum

## 사전 조건

Python 3.8 이상이 설치되어 있어야 합니다. 설치 지침은 Python 초급 가이드의 Python 다운로드 페이지를 참조하세요.

Python 버전 지원 매트릭스

| AWS CLI 버전 | 지원되는 Python 버전 |
| --- | --- |
| 1.32.0~현재 | Python 3.8 이상 |
| 1.27.0~1.31.x | Python 3.7 이상 |
| 1.20.0~1.26.x | Python 3.6 이상 |
| 1.19.0~1.19.x | Python 2.7 이상, Python 3.6 이상 |
| 1.17 ~ 1.18.x | Python 2.7 이상, Python 3.4 이상 |
| 1.0 ~ 1.16.x | Python 2.6 이상, Python 3.3 이상 |

# pip를 사용한 설치, 업데이트 또는 제거

대부분의 Amazon Linux 인스턴스는 pip를 사용하여 AWS CLI 버전 1을 사전 설치합니다.

## pip를 사용하여 Amazon Linux에 AWS CLI 버전 1 설치 또는 업데이트

현재 사용자에 대한 AWS CLI 최신 버전 1을 설치하려면 다음 지침을 사용합니다.

1. Python 버전 3 이상이 설치되어 있으면 `pip3`을 사용하는 것이 좋습니다. AWS CLI 버전 1의 최신 버전을 설치하거나 업데이트하는 `pip3 install` 데 사용합니다. Python 가상 환경(venv) 내에서 해당 명령을 실행할 경우, `--user` 옵션을 사용할 필요가 없습니다.

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

2. `aws`가 포함된 폴더가 `PATH` 변수의 일부인지 확인하세요.

   a. 사용자 디렉터리에서 셸의 프로파일 스크립트를 찾습니다. 어떤 셸을 가지고 있는지 잘 모르는 경우 `echo $SHELL`을 실행합니다.

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

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

   b. 다음 예제와 유사한 프로필 스크립트 끝에 내보내기 명령을 추가합니다.

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

   이 명령은 경로(이 예제에서 `$HOME/.local/bin`)를 기존 `$PATH` 변수 앞에 삽입합니다.

   c. 현재 세션에 프로필을 다시 로드하여 해당 변경 사항을 적용합니다.

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

3. 새 버전을 실행 중인지 확인하려면 `aws --version` 명령을 사용합니다.

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

### pip를 사용하여 AWS CLI 버전 1 제거

를 제거해야 하는 경우 를 AWS CLI사용합니다`pip uninstall`.

```
$ pip3 uninstall awscli
```

## yum을 사용한 설치, 업데이트 또는 제거

대부분의 Amazon Linux 2 인스턴스는 yum을 사용하여 AWS CLI 버전 1을 사전 설치합니다.

### yum을 사용하여 Amazon Linux에 AWS CLI 버전 1 설치 또는 업데이트

Amazon Linux에서 사용할 수 있는 AWS CLI 버전 1의 최신 버전을 설치하려면 다음 명령을 실행합니다.

```
$ sudo yum install awscli
```

Amazon Linux에서 사용할 수 있는 AWS CLI 버전 1의 최신 버전으로 업데이트하려면 다음 명령을 실행합니다.

```
$ sudo yum update awscli
```

최신 버전을 실행 중인지 확인하려면 `aws --version` 명령을 사용합니다.

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

### yum을 사용하여 AWS CLI 버전 1 제거

를 제거하려면 를 AWS CLI사용합니다`yum remove`.

```
$ sudo yum remove awscli
```

## AWS CLI 설치 및 제거 오류 문제 해결

를 설치하거나 제거한 후 문제가 발생하면 문제 해결 단계는 섹션을 AWS CLI참조오류 해결하세요. 가장 관련성이 높은 문제 해결 단계는 the section called “명령을 찾을 수 없음 오류”, the section

called "'aws --version' 명령이 설치한 버전과 다른 버전을 반환함" 및 the section called ""aws --version" 명령은 를 제거한 후 버전을 반환합니다. AWS CLI" 단원을 참조하세요.

# Linux에서 AWS CLI 버전 1 설치, 업데이트 및 제거

pip 패키지 관리자 또는 번들 설치 관리자를 사용하여 AWS Command Line Interface (AWS CLI) 버전 1과 대부분의 Linux 배포에 대한 종속성을 설치할 수 있습니다.

awscli 패키지는 apt 및 와 같은 다른 패키지 관리자의 리포지토리에서 사용할 수 있지만 에서 생성, 관리 또는 지원되지 yum않습니다 AWS. 이 안내서에 설명된 대로 공식 AWS 배포 지점에서 AWS CLI 만 를 설치하는 것이 좋습니다.

Sections



## 사전 조건

Python 3.8 이상이 설치되어 있어야 합니다. 설치 지침은 Python 초급 가이드의 Python 다운로드 페이지를 참조하세요.

Python 버전 지원 매트릭스

| AWS CLI 버전 | 지원되는 Python 버전 |
|---|---|
| 1.32.0~현재 | Python 3.8 이상 |
| 1.27.0~1.31.x | Python 3.7 이상 |
| 1.20.0~1.26.x | Python 3.6 이상 |
| 1.19.0~1.19.x | Python 2.7 이상, Python 3.6 이상 |

| AWS CLI 버전 | 지원되는 Python 버전 |
| --- | --- |
| 1.17 ~ 1.18.x | Python 2.7 이상, Python 3.4 이상 |
| 1.0 ~ 1.16.x | Python 2.6 이상, Python 3.3 이상 |

# 번들 설치 관리자를 사용하여 Linux에 AWS CLI 버전 1 설치 및 제거

Linux 또는 macOS에서는 번들 설치 관리자를 사용하여 AWS CLI의 버전 1을 설치할 수 있습니다. 번들 설치 관리자에는 모든 종속 항목이 포함되고 오프라인으로 사용할 수 있습니다.

> **Note**
>
> 번들 설치 관리자는 공백을 포함하는 경로에 설치하는 것을 지원하지 않습니다.

주제



## 와 함께 번들 설치 관리자를 사용하여 AWS CLI 버전 1 설치 **sudo**

다음 단계를 통해 Linux 또는 macOS 빌드의 명령줄에서 AWS CLI 버전 1을 설치할 수 있습니다.

다음은 단일 명령 집합으로 실행하기 위해 잘라내어 붙여 넣을 수 있는 아래에 설명된 설치 명령의 요약입니다.

최신 버전의 의 경우 다음 명령 블록을 AWS CLI 사용합니다.

```
$ 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
```

특정 버전의 의 경우 하이픈과 버전 번호를 파일 이름에 AWS CLI 추가합니다. 이 예제에서는 버전의 파일 이름 *1.16.312* 다음 명령이 `awscli-bundle-1.16.312.zip` 발생합니다.

```
$ 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
```

명령줄의 다음 단계에 따라 번들 설치 관리자를 사용하여 AWS CLI 버전 1을 설치합니다.

번들 설치 관리자를 사용하여 AWS CLI 버전 1을 설치하려면

1. 다음 방법 중 하나를 사용하여 AWS CLI 버전 1 번들 설치 관리자를 다운로드합니다.

   - `curl` 명령을 사용하여 다운로드합니다.

     최신 버전의 에는 다음 명령 블록을 AWS CLI 사용합니다.

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

     특정 버전의 의 경우 하이픈과 버전 번호를 파일 이름에 AWS CLI 추가합니다. 이 예제에서는 버전의 파일 이름 *1.16.312* 다음 명령이 `awscli-bundle-1.16.312.zip` 발생합니다.

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

   - 직접 링크를 사용하여 다운로드합니다.

     AWS CLI의 최신 버전인 경우: https://s3.amazonaws.com/aws-cli/awscli-bundle.zip

     특정 버전의 의 경우 하이픈과 버전 번호를 파일 이름에 AWS CLI추가합니다. 이 예제에서는 버전의 파일 이름 *1.16.312* 다음과 같은 URL이 `awscli-bundle-1.16.312.zip` 발생합니다. https://s3.amazonaws.com/aws-cli/awscli-bundle-1.16.312.zip

2. 패키지에서 파일을 추출합니다. 파일을 추출하기 위한 `unzip`이 없는 경우 Linux 배포의 내장된 패키지 관리자를 사용하여 설치하세요.

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

3. 설치 프로그램을 실행합니다. 설치 관리자는 AWS CLI 에 를 설치하고 `/usr/local/bin` 디렉터리에 symlink`aws`를 `/usr/local/aws` 생성합니다. `-b` 옵션을 사용하여 symlink를 생성하면 사용자의 `$PATH` 변수에 설치 디렉터리를 지정할 필요가 없습니다. 이렇게 하면 모든 사용자가 디렉터리`aws`에서 를 입력하여 AWS CLI 를 호출할 수 있습니다.

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

기본적으로 설치 스크립트는 시스템 기본 버전의 Python에서 실행됩니다. 대체 버전의 Python을 설치하고 해당 버전을 사용하여 를 설치하려는 경우 다음과 같이 Python 실행 파일에 대한 절대 경로를 통해 해당 버전으로 설치 스크립트를 AWS CLI실행합니다.

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

4. 이 올바르게 AWS CLI 설치되었는지 확인합니다.

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

오류가 발생한 경우 에 대한 오류 해결 AWS CLI 단원을 참조하세요.

## 번들 설치 관리자를 사용하여 AWS CLI 버전 1 설치 **sudo**

sudo 권한이 없거나 현재 사용자 AWS CLI 에 대해서만 를 설치하려는 경우 이전 명령의 수정된 버전을 사용할 수 있습니다. 처음 두 명령은 동일합니다.

최신 버전의 에는 다음 명령 블록을 AWS CLI 사용합니다.

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

특정 버전의 의 경우 하이픈과 버전 번호를 파일 이름에 AWS CLI 추가합니다. 이 예제에서는 버전의 파일 이름 *1.16.312* 다음 명령이 awscli-bundle-1.16.312.zip 발생합니다.

```
$ 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
```

현재 사용자에 대한 AWS CLI 버전 1을 설치하려면

1. 다음 방법 중 하나로 AWS CLI 버전 1 번들 설치 관리자를 다운로드합니다.

- `curl` 명령을 사용하여 다운로드합니다.

  최신 버전의 의 경우 다음 명령 블록을 AWS CLI 사용합니다.

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

  특정 버전의 의 경우 하이픈과 버전 번호를 파일 이름에 AWS CLI 추가합니다. 이 예제에서는 버전의 파일 이름 *1.16.312* 다음 명령이 `awscli-bundle-1.16.312.zip` 발생합니다.

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

- 직접 링크를 사용하여 다운로드합니다.

  AWS CLI의 최신 버전인 경우: https://s3.amazonaws.com/aws-cli/awscli-bundle.zip

  특정 버전의 의 경우 하이픈과 버전 번호를 파일 이름에 AWS CLI추가합니다. 이 예제에서는 버전의 파일 이름 *1.16.312* 다음과 같은 URL이 `awscli-bundle-1.16.312.zip` 발생합니다. https://s3.amazonaws.com/aws-cli/awscli-bundle-1.16.312.zip

2. `unzip`을 사용하여 패키지에서 파일을 추출합니다. `unzip`이 없는 경우 Linux 배포의 내장된 패키지 관리자를 사용하여 설치하세요.

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

3. 설치 프로그램을 실행합니다. 설치 관리자는 AWS CLI 에 를 설치하고 `/usr/local/bin` 디렉터리에 symlinkaws를 `/usr/local/aws` 생성합니다. 이 명령은 `-b` 파라미터를 사용하여 설치 관리자가 aws symlink 파일을 저장할 디렉터리를 지정합니다. 지정된 폴더에 대한 쓰기 권한이 있어야 합니다.

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

   이렇게 AWS CLI 하면 가 기본 위치(`~/.local/lib/aws`)에 설치되고 에 심볼 링크(심링크)가 생성됩니다`~/bin/aws`. symlink가 작동하려면 `~/bin`이 PATH 환경 변수에 있어야 합니다.

   ```
   $ 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. AWS CLI 버전 1이 PATH 변수의 일부인 디렉터리를 확인합니다.

   a. 사용자 폴더에서 셸의 프로파일 스크립트를 찾습니다. 어떤 셸을 가지고 있는지 잘 모르는 경우 `echo $SHELL`을 실행합니다.

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

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

   b. 다음 예제와 유사한 프로필 스크립트 끝에 내보내기 명령을 추가합니다.

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

   이 명령은 경로(이 예제에서 `~/.local/bin`)를 기존 PATH 변수 앞에 삽입합니다.

   c. 현재 세션에 프로필을 다시 로드하여 해당 변경 사항을 적용합니다.

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

5. 이 올바르게 AWS CLI 설치되었는지 확인합니다.

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

   오류가 발생한 경우 에 대한 오류 해결 AWS CLI 단원을 참조하세요.

## AWS CLI 버전 1 번들 설치 관리자 제거

1. 번들 설치 관리자를 AWS CLI 사용하여 를 설치한 경우 다음 지침을 따릅니다. 번들 설치 관리자는 선택적 symlink를 제외하고 설치 디렉터리 외부에 아무 것도 넣지 않으므로, 설치 제거는 이 두 항목을 삭제하는 것만큼 간단합니다.

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

2. (선택 사항) `.aws` 폴더에서 공유 AWS SDK 및 AWS CLI 설정 정보를 제거합니다.

> **Warning**
>
> 이러한 구성 및 자격 증명 설정은 모든 AWS SDKs 및 에서 공유됩니다 AWS CLI. 이 폴더를 제거하면 아직 시스템에 있는 AWS SDKs에서 액세스할 수 없습니다.

`.aws` 폴더의 기본 위치는 플랫폼마다 다르며, 기본적으로 폴더는 에 있습니다.*~/.aws/*. 사용자에게 이 디렉터리에 대한 쓰기 권한이 있는 경우 를 사용할 필요가 없습니다`sudo`.

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

# pip를 사용하여 AWS CLI 버전 1 설치 및 제거

주제



## PIP 설치

`pip`가 아직 설치되지 않은 경우 Python Packaging Authority에서 제공하는 스크립트를 사용하여 설치할 수 있습니다. `pip --version`을 실행하여 해당 버전의 Linux에 Python과 `pip`가 이미 포함되어 있는지 확인합니다. Python 버전 3 이상이 설치되어 있으면 `pip3` 명령을 사용하는 것이 좋습니다.

1. `curl` 명령을 사용하여 설치 스크립트를 다운로드합니다. 다음 명령은 `-O`(대문자 “O”) 파라미터를 사용하여 다운로드된 파일을 원격 호스트에서와 동일한 이름을 사용하여 현재 디렉터리에 저장하도록 지정합니다.

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

2. `python` 또는 `python3` 명령으로 스크립트를 실행하여 `pip` 및 기타 필요한 지원 패키지의 최신 버전을 다운로드하여 설치합니다. `--user` 스위치를 포함하면 스크립트는 `pip`를 `~/.local/bin` 경로에 설치합니다.

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

3. pip가 포함된 디렉터리가 PATH 변수의 일부인지 확인합니다.

   a. 사용자 폴더에서 셸의 프로파일 스크립트를 찾습니다. 어떤 셸을 가지고 있는지 잘 모르는 경우 `echo $SHELL`을 실행합니다.

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

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

   b. 다음 예제와 유사한 프로필 스크립트 끝에 내보내기 명령을 추가합니다.

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

   이 명령은 경로(이 예제에서 `~/.local/bin`)를 기존 PATH 변수 앞에 삽입합니다.

   c. 현재 세션에 프로필을 다시 로드하여 해당 변경 사항을 적용합니다.

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

4. pip 또는 pip3이 제대로 설치되었는지 확인하려면 다음 명령을 실행합니다.

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

## pip를 사용하여 AWS CLI 버전 1 설치 및 업데이트

1. pip 또는 pip3 명령을 사용하여 AWS CLI를 설치하거나 업데이트합니다. Python 버전 3 이상을 사용하는 경우에는 pip3 명령을 사용하는 것이 좋습니다. `--user` 스위치는 를 AWS CLI 에 pip 설치합니다`~/.local/bin`.

   최신 버전의 에는 다음 명령 블록을 AWS CLI 사용합니다.

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

   특정 버전의 의 경우 파일 이름에 2 = 기호와 버전 번호를 AWS CLI 추가합니다. = 이 예제에서는 버전의 파일 이름 *1.16.312* 는 *==1.16.312* 다음 명령이 발생합니다.

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

> **Note**
>
> 터미널에 해당하는 인용 규칙을 사용합니다. = 문자를 사용하려는 경우 제대로 이스케이프 처리하기 위해 작은따옴표 또는 큰따옴표를 사용해야 할 수 있습니다. 다음 예제에서는 작은따옴표를 사용하여 이스케이프 처리합니다.
>
> ```
> $ pip3 install 'awscli==1.16.312' --upgrade --user
> ```

2. 이 올바르게 AWS CLI 설치되었는지 확인합니다.

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

오류가 발생한 경우 에 대한 오류 해결 AWS CLI 단원을 참조하세요.

## pip를 AWS CLI 사용하여 제거

1. 를 사용하여 AWS CLI 버전 1을 설치한 경우 를 사용하여 도 제거pip해야 합니다pip.

```
$ pip uninstall awscli
```

Python 2 또는 3 버전을 사용하는 경우 pip2 또는 pip3 명령을 사용해야 할 수 있습니다. aws --version 명령을 사용하여 설치된 버전 1과 연결된 Python AWS CLI 버전을 확인합니다.

```
$ pip3 uninstall awscli
```

모든 파일을 제거하려면 명령 프롬프트 창이나 컴퓨터를 다시 시작해야 할 수 있습니다.

2. (선택 사항) .aws 폴더에서 공유 AWS SDK 및 AWS CLI 설정 정보를 제거합니다.

> **Warning**
>
> 이러한 구성 및 자격 증명 설정은 모든 AWS SDKs 및 에서 공유됩니다 AWS CLI. 이 폴더를 제거하면 아직 시스템에 있는 AWS SDKs에서 액세스할 수 없습니다.

.aws 폴더의 기본 위치는 플랫폼마다 다르며, 기본적으로 폴더는 에 있습니다.*~/.aws/*. 사용자에게 이 디렉터리에 대한 쓰기 권한이 있는 경우 를 사용할 필요가 없습니다sudo.

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

## Snapcraft를 사용하여 AWS CLI 버전 1 설치 및 제거

주제

- 스냅 설치
- 스냅을 사용하여 AWS CLI 버전 1 설치 및 업데이트
- 스냅을 AWS CLI 사용하여 제거

### 스냅 설치

아직 snap 설치하지 않은 경우 Canonical Snapcraft에서 제공하는 지침을 사용하여 설치할 수 있습니다. 를 실행snap version하여 Linux 버전에 가 이미 포함되어 있는지 확인합니다snap.

1. 플랫폼에 Snapcraft를 설치합니다. Snapcraft 설치에 대한 자세한 내용은 Snap 설명서의 데몬 설치를 참조하세요.
2. PATH 변수가 올바르게 업데이트되도록 시스템을 다시 시작합니다. 설치 문제가 있는 경우 스냅 설명서 의 일반적인 문제 해결의 단계를 따릅니다.
3. snap 이 올바르게 설치되었는지 확인하려면 다음 명령을 실행합니다.

```
$ snap version
```

### 스냅을 사용하여 AWS CLI 버전 1 설치 및 업데이트

1. AWS CLI 버전 1에 대해 다음 snap install 명령을 실행합니다.

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

권한에 따라 명령에 를 추가해야 sudo 할 수 있습니다.

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

2. 이 올바르게 AWS CLI 설치되었는지 확인합니다.

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

오류가 발생한 경우 에 대한 오류 해결 AWS CLI 단원을 참조하세요.

### 스냅을 AWS CLI 사용하여 제거

1. 를 사용하여 AWS CLI 버전 1을 설치한 경우 를 사용하여 제거해야 snap합니다snap.

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

모든 파일을 제거하려면 명령 프롬프트 창이나 컴퓨터를 다시 시작해야 할 수 있습니다.

2. (선택 사항) .aws 폴더에서 공유 AWS SDK 및 AWS CLI 설정 정보를 제거합니다.

> **Warning**
>
> 이러한 구성 및 자격 증명 설정은 모든 AWS SDKs 및 에서 공유됩니다 AWS CLI. 이 폴더를 제거하면 아직 시스템에 있는 AWS SDKs에서 액세스할 수 없습니다.

.aws 폴더의 기본 위치는 플랫폼마다 다르며, 기본적으로 폴더는 에 있습니다.*~/.aws/*. 이 디렉터리에 대한 쓰기 권한이 있는 경우 를 사용할 필요가 없습니다sudo.

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

## 명령줄 경로에 AWS CLI 버전 1 실행 파일 추가

pip 또는 를 사용하여 를 설치snap한 후 aws 실행 파일을 운영 체제의 PATH 환경 변수에 추가해야 할 수 있습니다.

다음 명령을 실행하여 AWS CLI 에 pip 설치된 폴더를 확인할 수 있습니다.

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

이를 ~/.local/bin/이라고 할 수 있는데, Linux에서 /home/username은 ~에 해당하기 때문입니다.

--user 스위치를 생략하여 사용자 모드에서 설치하지 않았다면 실행 파일이 Python의 bin 폴더에 있을 수 있습니다. Python 설치 위치를 모르는 경우, 다음 명령을 실행하세요.

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

실제 실행 파일이 아니라 symlink 경로가 출력될 수 있습니다. ls -al을 실행하여 어디를 가리키는지 확인합니다.

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

pip는 Python 애플리케이션이 있는 것과 동일한 폴더에 프로그램을 설치합니다. 이 폴더를 PATH 변수에 추가합니다.

**PATH 변수를 수정하려면**

1. 사용자 디렉터리에서 셸의 프로파일 스크립트를 찾습니다. 어떤 셸을 가지고 있는지 잘 모르는 경우 echo $SHELL을 실행합니다.

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

   - Bash - .bash_profile, .profile 또는 .bash_login
   - Zsh – .zshrc
   - Tcsh - .tcshrc, .cshrc 또는 .login
2. 내보내기 명령을 프로필 스크립트에 추가하세요.

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

   이 명령은 이 예제의 ~/.local/bin 경로를 현재 PATH 변수에 추가합니다.
3. 현재 세션에 업데이트된 프로필을 로드합니다.

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

## AWS CLI 설치 및 제거 오류 문제 해결

를 설치하거나 제거한 후 문제가 발생하면 문제 해결 단계는 섹션을 AWS CLI참조오류 해결하세요. 가장 관련성이 높은 문제 해결 단계는 the section called “명령을 찾을 수 없음 오류”, the section called “'aws --version' 명령이 설치한 버전과 다른 버전을 반환함” 및 the section called ““aws --version” 명령은 를 제거한 후 버전을 반환합니다. AWS CLI” 단원을 참조하세요.

# macOS에서 AWS CLI 버전 1 설치, 업데이트 및 제거

번들 설치 관리자 또는 를 사용하여 AWS Command Line Interface (AWS CLI) 버전 1과 해당 종속 항목을 macOS에 설치할 수 있습니다pip.

Sections



## 사전 조건

macOS 에 AWS CLI 버전 1을 설치하려면 먼저 Python 3.8 이상이 설치되어 있어야 합니다. 설치 지침은 Python 초급 가이드의 Python 다운로드 페이지를 참조하세요.

Python 버전 지원 매트릭스

| AWS CLI 버전 | 지원되는 Python 버전 |
| --- | --- |
| 1.32.0~현재 | Python 3.8 이상 |
| 1.27.0~1.31.x | Python 3.7 이상 |
| 1.20.0~1.26.x | Python 3.6 이상 |
| 1.19.0~1.19.x | Python 2.7 이상, Python 3.6 이상 |

| AWS CLI 버전 | 지원되는 Python 버전 |
| --- | --- |
| 1.17 ~ 1.18.x | Python 2.7 이상, Python 3.4 이상 |
| 1.0 ~ 1.16.x | Python 2.6 이상, Python 3.3 이상 |

# 번들 설치 관리자를 사용하여 macOS에서 AWS CLI 버전 1 설치, 업데이트 및 제거

Linux 또는 macOS에서는 번들 설치 관리자를 사용하여 AWS Command Line Interface (AWS CLI)의 버전 1을 설치할 수 있습니다. 번들 설치 관리자에는 모든 종속 항목이 포함되고 오프라인으로 사용할 수 있습니다.

번들 설치 관리자는 공백을 포함하는 경로에 설치하는 것을 지원하지 않습니다.

주제



## 를 사용하여 번들 설치 관리자를 사용하여 AWS CLI 버전 1 설치 **sudo**

다음 단계를 통해 모든 macOS 빌드의 명령줄에서 AWS CLI 버전 1을 설치할 수 있습니다.

다음에 단일 명령 집합으로 실행하기 위해 잘라내어 붙여 넣을 수 있는 설치 명령이 요약되어 있습니다.

최신 버전의 에는 다음 명령 블록을 AWS CLI 사용합니다.

```
$ 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
```

특정 버전의 의 경우 하이픈과 버전 번호를 파일 이름에 AWS CLI 추가합니다. 이 예제에서는 버전의 파일 이름 *1.16.312* 다음 명령이 `awscli-bundle-1.16.312.zip` 발생합니다.

```
$ 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
```

번들 설치 관리자를 사용하여 AWS CLI 버전 1을 설치하려면

1. 다음 방법 중 하나로 AWS CLI 버전 1 번들 설치 관리자를 다운로드합니다.

   - `curl` 명령을 사용하여 다운로드합니다.

     최신 버전의 AWS CLI의 경우 다음 명령 블록을 사용하세요.

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

     특정 버전의 AWS CLI의 경우 파일 이름에 하이픈과 버전 번호를 추가하세요. 이 예제에서는 버전의 파일 이름 *1.16.312* 다음 명령이 `awscli-bundle-1.16.312.zip` 발생합니다.

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

   - 직접 링크를 사용하여 다운로드합니다.

     AWS CLI의 최신 버전인 경우: https://s3.amazonaws.com/aws-cli/awscli-bundle.zip

     특정 버전의 의 경우 하이픈과 버전 번호를 파일 이름에 AWS CLI추가합니다. 이 예제에서는 버전의 파일 이름 *1.16.312* 다음과 같은 URL이 `awscli-bundle-1.16.312.zip` 발생합니다. https://s3.amazonaws.com/aws-cli/awscli-bundle-1.16.312.zip

2. 패키지에서 파일을 추출(압축 해제)합니다. 가 없는 경우 macOs 배포의 내장 패키지 관리자를 `unzip`사용하여 설치합니다.

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

3. 설치 프로그램을 실행합니다. 설치 프로그램은 AWS CLI 에 를 설치하고 `/usr/local/bin` 폴더에 symlink`aws`를 `/usr/local/aws` 생성합니다. `-b` 옵션을 사용하여 symlink를 생성하면 사용자의 `$PATH` 변수에 설치 폴더를 지정할 필요가 없습니다. 이렇게 하면 모든 사용자가 디렉터리`aws`에서 를 입력하여 AWS CLI 를 호출할 수 있습니다.

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

기본적으로 설치 스크립트는 시스템 기본 버전의 Python에서 실행됩니다. 대체 버전의 Python을 설치하고 이를 사용하여 를 설치하려는 경우 다음과 같이 Python 실행 파일에 대한 절대 경로를 통해 해당 버전으로 설치 스크립트를 AWS CLI실행합니다.

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

4. 이 올바르게 AWS CLI 설치되었는지 확인합니다.

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

오류가 발생한 경우 에 대한 오류 해결 AWS CLI 단원을 참조하세요.

## 번들 설치 관리자를 사용하여 AWS CLI 버전 1 설치 **sudo**

sudo 권한이 없거나 현재 사용자 AWS CLI 에 대해서만 를 설치하려는 경우 이전 명령의 수정된 버전을 사용할 수 있습니다. 처음 두 명령은 동일합니다.

최신 버전의 에는 다음 명령 블록을 AWS CLI 사용합니다.

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

특정 버전의 의 경우 하이픈과 버전 번호를 파일 이름에 AWS CLI 추가합니다. 이 예제에서는 버전의 파일 이름 *1.16.312* 다음 명령이 awscli-bundle-1.16.312.zip 발생합니다.

```
$ 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
```

현재 사용자에 대한 AWS CLI 버전 1을 설치하려면

1. 다음 방법 중 하나를 사용하여 AWS CLI 버전 1 번들 설치 관리자를 다운로드합니다.

   - curl 명령을 사용하여 다운로드합니다.

최신 버전의 AWS CLI의 경우 다음 명령 블록을 사용하세요.

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

특정 버전의 AWS CLI의 경우 파일 이름에 하이픈과 버전 번호를 추가하세요. 이 예제에서는 버전의 파일 이름 *1.16.312* 다음 명령이 awscli-bundle-1.16.312.zip 발생합니다.

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

- 직접 링크를 사용하여 다운로드합니다.

  AWS CLI의 최신 버전인 경우: https://s3.amazonaws.com/aws-cli/awscli-bundle.zip

  특정 버전의 의 경우 하이픈과 버전 번호를 파일 이름에 AWS CLI추가합니다. 이 예제에서는 버전의 파일 이름 *1.16.312* 다음과 같은 URL이 awscli-bundle-1.16.312.zip 발생합니다. https://s3.amazonaws.com/aws-cli/awscli-bundle-1.16.312.zip

2. 패키지에서 파일을 추출합니다. unzip이 없는 경우 Linux 배포의 내장된 패키지 관리자를 사용하여 설치하세요.

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

3. 설치 프로그램을 실행합니다. 설치 프로그램은 AWS CLI 에 를 설치하고 /usr/local/bin 디렉터리에 symlinkaws를 /usr/local/aws 생성합니다. 이 명령은 -b 파라미터를 사용하여 설치 관리자가 aws symlink 파일을 저장할 디렉터리를 지정합니다. 지정된 디렉터리에 대한 쓰기 권한이 있어야 합니다.

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

   이렇게 AWS CLI 하면 가 기본 위치(~/.local/lib/aws)에 설치되고 에 심볼 링크(심볼 링크)가 생성됩니다~/bin/aws. symlink가 작동하려면 ~/bin이 $PATH 환경 변수에 있어야 합니다.

   ```
   $ 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. AWS CLI 버전 1이 설치된 폴더가 $PATH 변수의 일부인지 확인합니다.

a. 사용자 폴더에서 셸의 프로파일 스크립트를 찾습니다. 어떤 셸을 가지고 있는지 잘 모르는 경우 `echo $SHELL`을 실행합니다.

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

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

b. 다음 예제와 유사한 프로필 스크립트 끝에 내보내기 명령을 추가합니다.

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

이 명령은 경로(이 예제에서 `~/.local/bin`)를 기존 `PATH` 변수 앞에 삽입합니다.

c. 현재 세션에 프로필을 다시 로드하여 해당 변경 사항을 적용합니다.

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

5. 이 올바르게 AWS CLI 설치되었는지 확인합니다.

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

오류가 발생한 경우 에 대한 오류 해결 AWS CLI 단원을 참조하세요.

## AWS CLI 버전 1 번들 설치 관리자 제거

1. 번들 설치 관리자는 선택적 symlink를 제외한 모든 것을 설치 디렉터리 안에 넣으므로 제거하려면 두 항목을 삭제하면 됩니다.

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

2. (선택 사항) `.aws` 폴더에서 공유 AWS SDK 및 AWS CLI 설정 정보를 제거합니다.

> **Warning**
>
> 이러한 구성 및 자격 증명 설정은 모든 AWS SDKs 및 에서 공유됩니다 AWS CLI. 이 폴더를 제거하면 아직 시스템에 있는 AWS SDKs에서 액세스할 수 없습니다.

`.aws` 폴더의 기본 위치는 플랫폼마다 다르며 기본적으로 폴더는 에 있습니다.*~/.aws/*. 사용자에게 이 디렉터리에 대한 쓰기 권한이 있는 경우 를 사용할 필요가 없습니다`sudo`.

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

# pip를 사용하여 AWS CLI 버전 1 설치, 업데이트 및 제거

`pip`를 직접 사용하여 AWS CLI를 설치할 수 있습니다.

주제

- PIP 설치
- pip를 AWS CLI 사용하여 설치 및 업데이트
- AWS CLI 버전 1 실행 파일을 macOS 명령줄 경로에 추가합니다.
- pip를 AWS CLI 사용하여 제거

## PIP 설치

`pip`가 아직 설치되지 않은 경우 Python Packaging Authority에서 제공하는 스크립트를 사용하여 설치할 수 있습니다. `pip --version`을 실행하여 해당 버전의 Linux에 Python과 `pip`가 이미 포함되어 있는지 확인합니다. Python 버전 3 이상이 설치되어 있으면 `pip3` 명령을 사용하는 것이 좋습니다.

1. `curl` 명령을 사용하여 설치 스크립트를 다운로드합니다. 다음 명령은 `-O`(대문자 “O”) 파라미터를 사용하여 다운로드된 파일을 원격 호스트에서와 동일한 이름을 사용하여 현재 폴더에 저장하도록 지정합니다.

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

2. `python` 또는 `python3` 명령으로 스크립트를 실행하여 `pip` 및 기타 필요한 지원 패키지의 최신 버전을 다운로드하여 설치합니다. `--user` 스위치를 포함하면 스크립트는 `pip`를 `~/.local/bin` 경로에 설치합니다.

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

## pip를 AWS CLI 사용하여 설치 및 업데이트

1. pip 또는 pip3 명령을 사용하여 AWS CLI를 설치합니다. Python 버전 3 이상을 사용하는 경우에는 pip3 명령을 사용하는 것이 좋습니다.

   최신 버전의 에는 다음 명령 블록을 AWS CLI 사용합니다.

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

   의 특정 버전에 대해 2 = 기호와 버전 번호를 파일 이름에 AWS CLI 추가합니다. = 이 예제에서는 버전의 파일 이름 *1.16.312* 가 될 것입니다.*==1.16.312* 그러면 다음 명령이 생성됩니다.

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

   > **Note**
   >
   > 터미널에 해당하는 인용 규칙을 사용합니다. = 문자를 사용하려는 경우 제대로 이스케이프 처리하기 위해 작은따옴표 또는 큰따옴표를 사용해야 할 수 있습니다. 다음 예제에서는 작은따옴표를 사용하여 이스케이프 처리합니다.
   >
   > ```
   > $ pip3 install 'awscli==1.16.312' --upgrade --user
   > ```

2. 이 올바르게 설치 AWS CLI 되었는지 확인합니다.

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

   프로그램을 찾을 수 없는 경우 프로그램을 명령줄 경로에 추가합니다.

## AWS CLI 버전 1 실행 파일을 macOS 명령줄 경로에 추가합니다.

pip를 사용하여 설치한 후 aws 프로그램을 운영 체제의 PATH 환경 변수에 추가해야 할 수 있습니다. 프로그램의 위치는 Python 설치 위치에 따라 달라집니다.

Example AWS CLI 설치 위치 - macOS with Python 3.6 및 **pip** (사용자 모드)

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

앞의 예제에 나온 버전을 현재 가지고 있는 Python 버전으로 대체합니다.

Python 설치 위치를 모르는 경우, `which python`을 실행하세요.

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

실제 프로그램이 아니라 symlink 경로가 출력될 수 있습니다. `ls -al`을 실행하여 어디를 가리키는지 확인합니다.

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

`pip`는 Python 애플리케이션이 있는 것과 동일한 폴더에 프로그램을 설치합니다. 이 폴더를 `PATH` 변수에 추가합니다.

**`PATH`** 변수를 수정하려면

1. 사용자 디렉터리에서 셸의 프로파일 스크립트를 찾습니다. 어떤 셸을 가지고 있는지 잘 모르는 경우 `echo $SHELL`을 실행합니다.

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

   - Bash - `.bash_profile`, `.profile` 또는 `.bash_login`
   - Zsh – `.zshrc`
   - Tcsh - `.tcshrc`, `.cshrc` 또는 `.login`
2. 내보내기 명령을 프로필 스크립트에 추가하세요.

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

   이 명령은 이 예제의 `~/.local/bin` 경로를 현재 `PATH` 변수에 추가합니다.
3. 현재 세션에 업데이트된 프로필을 로드합니다.

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

### pip를 AWS CLI 사용하여 제거

1. 를 사용하여 AWS CLI 버전 1을 설치한 경우 를 사용하여 제거해야 pip합니다pip.

   ```
   $ pip uninstall awscli
   ```

   Python 2 또는 3 버전을 사용하는 경우 pip2 또는 pip3 명령을 사용해야 할 수 있습니다. aws --version 명령을 사용하여 설치된 버전 1과 연결된 Python AWS CLI 버전을 확인합니다.

   ```
   $ pip3 uninstall awscli
   ```

   모든 파일을 제거하려면 명령 프롬프트 창이나 컴퓨터를 다시 시작해야 할 수 있습니다.

2. (선택 사항) .aws 폴더에서 공유 AWS SDK 및 AWS CLI 설정 정보를 제거합니다.

   > **Warning**
   >
   > 이러한 구성 및 자격 증명 설정은 모든 AWS SDKs 및 에서 공유됩니다 AWS CLI. 이 폴더를 제거하면 아직 시스템에 있는 AWS SDKs에서 액세스할 수 없습니다.

   .aws 폴더의 기본 위치는 플랫폼마다 다르며 기본적으로 폴더는 에 있습니다.*~/.aws/*. 사용자에게 이 디렉터리에 대한 쓰기 권한이 있는 경우 를 사용할 필요가 없습니다sudo.

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

## AWS CLI 설치 및 제거 오류 문제 해결

를 설치하거나 제거한 후 문제가 발생하면 문제 해결 단계는 섹션을 AWS CLI참조오류 해결하세요. 가장 관련성이 높은 문제 해결 단계는 the section called “명령을 찾을 수 없음 오류”, the section called “'aws --version' 명령이 설치한 버전과 다른 버전을 반환함” 및 the section called ““aws --version” 명령은 를 제거한 후 버전을 반환합니다. AWS CLI” 단원을 참조하세요.

# Windows에서 AWS CLI 버전 1 설치, 업데이트 및 제거

독립형 설치 관리자 AWS Command Line Interface (권장AWS CLI) 또는 Python의 패키지 관리자pip인 를 사용하여 Windows에 () 버전 1을 설치할 수 있습니다.

명령을 입력할 때 프롬프트 기호(`C:\>`)를 포함시키지 마십시오. 이 기호는 입력하는 명령을 AWS CLI에서 반환되는 출력과 구별하기 위해 프로그램 목록에 포함되어 있습니다. 이 설명서의 나머지 부분에서는 명령이 Windows에 특정한 경우를 제외하고 일반 프롬프트 기호($)를 사용합니다.

주제



## MSI 설치 관리자를 사용하여 AWS CLI 버전 1 설치, 업데이트 및 제거

AWS CLI 버전 1은 Windows XP 이상에서 지원됩니다. Windows 사용자의 경우 MSI 설치 패키지는 다른 사전 요구 사항을 설치하지 않고 AWS CLI 버전 1을 설치하는 친숙하고 편리한 방법을 제공합니다.

### MSI 설치 관리자를 사용하여 AWS CLI 버전 1 설치 및 업데이트

의 릴리스 페이지를 확인하여 최신 버전이 릴리스된 시기를 GitHub 확인합니다. 업데이트가 릴리스되면 설치 프로세스를 반복하여 최신 버전의 AWS CLI 버전 1을 가져와야 합니다.

1. 적절한 MSI 설치 관리자를 다운로드합니다.

   - AWS CLI MSI Windows용 설치 관리자(64비트): https://s3.amazonaws.com/aws-cli/AWSCLI64PY3.msi
   - AWS CLI MSI Windows용 설치 관리자(32비트): https://s3.amazonaws.com/aws-cli/AWSCLI32PY3.msi
   - AWS CLI Windows용 통합 설정 파일: https://s3.amazonaws.com/aws-cli/AWSCLISetup.exe(32비트 및 64비트 MSI 설치 관리자 모두 포함, 올바른 버전 자동 설치)
2. 다운로드한 MSI 설치 관리자 또는 설정 파일을 실행합니다.
3. 화면에 표시되는 지시 사항을 따릅니다. 기본적으로 AWS CLI 버전 1은 `C:\Program Files\Amazon\AWSCLI` (64비트 버전) 또는 `C:\Program Files (x86)\Amazon\AWSCLI` (32비트 버전)에 설치됩니다.

4. 설치를 확인하려면 명령 프롬프트에서 `aws --version` 명령을 사용합니다. 시작 메뉴를 열고 `cmd`를 검색하여 명령 프롬프트를 시작할 수 있습니다.

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

Windows에서 프로그램을 찾을 수 없는 경우 명령 프롬프트를 닫았다가 다시 열어 경로를 새로 고치거나 설치 디렉터리를 환경 변수에 수동으로 추가해야 할 수 있습니다PATH.

## AWS CLI 버전 1 제거

다음 제거 지침을 사용하려면 MSI 설치 관리자 또는 설정 파일과 함께 AWS CLI 버전 1을 설치해야 합니다.

1. 다음 중 하나를 수행하여 Programs and Features(프로그램 및 기능)를 엽니다.

   - Control Panel(제어판)을 연 후 Programs and Features(프로그램 및 기능)를 선택합니다.
   - 명령 프롬프트를 열고 다음 명령을 입력합니다.

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

2. AWS Command Line Interface라는 항목을 선택한 다음, Uninstall(제거)을 선택하여 제거 프로그램을 시작합니다.
3. 를 제거하려는지 확인합니다 AWS CLI.
4. (선택 사항) `.aws` 폴더에서 공유 AWS SDK 및 AWS CLI 설정 정보를 제거합니다.

> **Warning**
>
> 이러한 구성 및 자격 증명 설정은 모든 AWS SDKs 및 에서 공유됩니다 AWS CLI. 이 폴더를 제거하면 아직 시스템에 있는 AWS SDKs에서 액세스할 수 없습니다.

`.aws` 폴더의 기본 위치는 플랫폼마다 다르며, 기본적으로 폴더는 에 있습니다.*%UserProfile%\.aws*.

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

# Windows에서 Python 및 pip를 사용하여 AWS CLI 버전 1 설치, 업데이트 및 제거

Python Software Foundation은 pip가 포함된 Windows용 설치 관리자를 제공합니다.

## 사전 조건

Python 3.8 이상이 설치되어 있어야 합니다. 설치 지침은 Python 초급 가이드의 Python 다운로드 페이지를 참조하세요.

## pip를 사용하여 AWS CLI 버전 1 설치 및 업데이트

1. AWS CLI 버전 1을 설치하려면 pip3 명령(Python 버전 3 이상을 사용하는 경우) 또는 pip 명령을 사용합니다.

   최신 버전의 의 경우 다음 명령 블록을 AWS CLI 사용합니다.

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

   특정 버전의 의 경우 보다 작은 기호와 버전 번호를 파일 이름에 AWS CLI 추가합니다. < 이 예제에서는 버전의 파일 이름 *1.16.312* 는 *<1.16.312* 다음 명령이 발생합니다.

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

2. AWS CLI 버전 1이 올바르게 설치되었는지 확인합니다. 응답이 없으면 명령줄 경로에 AWS CLI 버전 1 실행 파일 추가 섹션을 참조하세요.

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

## pip를 사용하여 AWS CLI 버전 1 제거

1. 를 사용하여 AWS CLI 버전 1을 설치한 경우 를 사용하여 도 제거pip해야 합니다pip.

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

   Python 2 또는 3 버전을 사용하는 경우 pip2 또는 pip3 명령을 사용해야 할 수 있습니다. aws --version 명령을 사용하여 설치된 버전 1과 연결된 Python AWS CLI 버전을 확인합니다.

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

모든 파일을 제거하려면 명령 프롬프트 창이나 컴퓨터를 다시 시작해야 할 수 있습니다.

2. (선택 사항) `.aws` 폴더에서 공유 AWS SDK 및 AWS CLI 설정 정보를 제거합니다.

> **Warning**
>
> 이러한 구성 및 자격 증명 설정은 모든 AWS SDKs 및 에서 공유됩니다 AWS CLI. 이 폴더를 제거하면 아직 시스템에 있는 AWS SDKs에서 액세스할 수 없습니다.

`.aws` 폴더의 기본 위치는 플랫폼마다 다르며, 기본적으로 폴더는 에 있습니다.*%UserProfile%\.aws*.

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

## 명령줄 경로에 AWS CLI 버전 1 실행 파일 추가

를 사용하여 AWS CLI 버전 1을 설치한 후 운영 체제의 PATH 환경 변수에 aws 프로그램을 pip추가합니다. MSI 설치 시 자동으로 발생합니다. 그러나 설치 후 aws 명령이 실행되지 않으면 수동으로 설정해야 할 수 있습니다.

1. where 명령을 사용하여 aws 파일 위치를 찾습니다. 기본적으로 where 명령은 시스템의 PATH에서 지정된 프로그램이 있는 위치를 표시합니다.

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

표시되는 경로는 플랫폼과 AWS CLI를 설치하는 데 사용한 방법에 따라 달라집니다. 버전 번호를 포함한 폴더 이름은 달라질 수 있습니다. 이러한 예제는 Python 버전 3.7 사용을 반영합니다. 필요에 따라 버전을 사용 중인 버전 번호로 바꿉니다. 일반적인 경로는 다음과 같습니다.

- Python 3 및 **pip3** – `C:\Program Files\Python37\Scripts\`
- Python 3 및 **pip3** --이전 버전의 Windows에서 사용자 옵션 - `%USERPROFILE%\AppData\Local\Programs\Python\Python37\Scripts`

- Python 3 및 **pip3** -- Windows 10에서의 사용자 옵션 - `%USERPROFILE%\AppData\Roaming\Python\Python37\Scripts`
- MSI 설치 프로그램(64비트) - `C:\Program Files\Amazon\AWSCLI\bin`
- MSI 설치 프로그램(32비트) - `C:\Program Files (x86)\Amazon\AWSCLI\bin`

파일 경로가 반환되는지 여부에 따라 다음 단계를 사용합니다.

A file path is returned

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

다음 명령을 실행하여 aws 프로그램이 어디에 설치되어 있는지 확인할 수 있습니다.

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

A file path is NOT returned

`where` 명령이 다음 오류를 반환하면 프로그램이 시스템 PATH에 없으며 이름을 입력하여 실행할 수 없습니다.

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

이 경우 `where` 파라미터와 함께 `/R` *`path`* 명령을 실행하여 모든 폴더를 검색하도록 한 후 경로를 수동으로 추가합니다. 명령줄 또는 파일 탐색기를 사용하여 프로그램이 컴퓨터에서 어디에 설치되어 있는지 검색합니다.

```
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. Windows 키를 누르고 **environment variables**를 입력하세요.
3. 계정의 환경 변수 편집을 선택합니다.

4. PATH를 선택한 다음 편집을 선택합니다.
5. 찾은 경로를 변수 값 필드에 추가합니다(예: ***C:\Program Files\Amazon\AWSCLI\bin\aws.exe***).
6. 확인을 두 번 선택하여 새 설정을 적용합니다.
7. 실행 중인 명령 프롬프트를 모두 닫았다가 명령 프롬프트 창을 다시 엽니다.

## AWS CLI 설치 및 제거 오류 문제 해결

를 설치하거나 제거한 후 문제가 발생하면 문제 해결 단계는 섹션을 AWS CLI참조오류 해결하세요. 가장 관련성이 높은 문제 해결 단계는 the section called “명령을 찾을 수 없음 오류”, the section called “'aws --version' 명령이 설치한 버전과 다른 버전을 반환함” 및 the section called ““aws --version” 명령은 를 제거한 후 버전을 반환합니다. AWS CLI” 단원을 참조하세요.

# 가상 환경에서 AWS CLI 버전 1 설치 및 업데이트

가상 환경에 AWS Command Line Interface (AWS CLI) 버전 1을 설치하여 요구 사항 버전이 다른 `pip` 패키지와 충돌하지 않도록 할 수 있습니다.

주제



## 사전 조건

- Python 3.8 이상 설치 지침은 Python 초급 가이드의 Python 다운로드 페이지를 참조하세요.

  Python 버전 지원 매트릭스

| AWS CLI 버전 | 지원되는 Python 버전 |
| --- | --- |
| 1.32.0~현재 | Python 3.8 이상 |
| 1.27.0~1.31.x | Python 3.7 이상 |
| 1.20.0~1.26.x | Python 3.6 이상 |

| AWS CLI 버전 | 지원되는 Python 버전 |
| --- | --- |
| 1.19.0~1.19.x | Python 2.7 이상, Python 3.6 이상 |
| 1.17 ~ 1.18.x | Python 2.7 이상, Python 3.4 이상 |
| 1.0 ~ 1.16.x | Python 2.6 이상, Python 3.3 이상 |

- `pip` 또는 `pip3`이 설치되어 있습니다.

## 가상 환경에서 AWS CLI 버전 1 설치 및 업데이트

1. `pip`를 사용하여 `virtualenv`를 설치합니다.

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

2. 가상 환경을 생성하고 이름을 지정합니다.

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

또는 `-p` 옵션을 사용하여 기본 버전 이외의 Python 버전을 지정할 수 있습니다.

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

3. 새 가상 환경을 활성화합니다.

Linux 또는 macOS

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

Windows

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

프롬프트가 변경되어 가상 환경이 활성임을 보여줍니다.

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

4. AWS CLI 버전 1을 가상 환경에 설치하거나 업데이트합니다.

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

5. AWS CLI 버전 1이 올바르게 설치되었는지 확인합니다.

```
$ 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. `deactivate` 명령을 사용하여 가상 환경을 종료할 수 있습니다. 새 세션을 시작할 때마다 환경을 다시 활성화해야 합니다.

## AWS CLI 설치 및 제거 오류 문제 해결

를 설치하거나 제거한 후 문제가 발생하면 문제 해결 단계는 섹션을 AWS CLI참조오류 해결하세요. 가장 관련성이 높은 문제 해결 단계는 the section called “명령을 찾을 수 없음 오류”, the section called “'aws --version' 명령이 설치한 버전과 다른 버전을 반환함” 및 the section called ““aws --version” 명령은 를 제거한 후 버전을 반환합니다. AWS CLI” 단원을 참조하세요.

# 에 대한 설정 구성 AWS CLI

이 섹션에서는 AWS Command Line Interface (AWS CLI)가 와 상호 작용하는 데 사용하는 설정을 구성하는 방법을 설명합니다 AWS. 여기에는 다음이 포함됩니다.

- 자격 증명은 를 호출하는 사용자를 식별합니다API. 액세스 자격 증명은 AWS 서버에 대한 요청을 암호화하여 자격 증명을 확인하고 관련 권한 정책을 검색하는 데 사용됩니다. 이러한 권한에 따라 수행할 수 있는 작업이 결정됩니다. 보안 인증 설정에 대한 자세한 내용은 인증 및 액세스 보안 인증 단원을 참조하세요.
- 기본 출력 형식 및 기본 리전과 같이 요청을 처리하는 AWS CLI 방법을 알려주는 AWS 기타 구성 세부 정보입니다.

> **Note**
>
> AWS 에서는 모든 수신 요청에 암호화 서명을 해야 합니다. 에서 이 AWS CLI 작업을 수행합니다. '서명'에는 AWS 서비스 AWS 에서 date/time stamp. Therefore, you must ensure that your computer's date and time are set correctly. If you don't, and the date/time in the signature is too far off of the date/time 인식한 가 포함되어 요청을 거부합니다.

## 구성 및 보안 인증 우선 순위

자격 증명 및 구성 설정은 시스템 또는 사용자 환경 변수, 로컬 AWS 구성 파일 또는 명령줄에 파라미터로 명시적으로 선언되는 등 여러 위치에 있습니다. 특정 위치가 다른 위치보다 우선합니다. AWS CLI 자격 증명 및 구성 설정이 다음 순서로 우선합니다.

1. 명령줄 옵션 - `--region`, `--output`, `--profile`와 같은 다른 위치의 설정을 재정의합니다.
2. 환경 변수 - 시스템의 환경 변수에 값을 저장할 수 있습니다.
3. 역할 수임 - 구성 또는 `aws sts assume-role` 명령을 통해 IAM 역할의 권한을 수임합니다.
4. 웹 ID로 역할 수임 - 구성 또는 `aws sts assume-role` 명령을 통해 웹 ID를 사용하여 IAM 역할의 권한을 수임합니다.
5. 보안 인증 파일 - `aws configure` 명령을 실행하면 `credentials` 및 `config` 파일이 업데이트됩니다. `credentials` 파일은 `~/.aws/credentials`(Linux 또는 macOS) 또는 `C:\Users\`*`USERNAME`*`\.aws\credentials`(Windows)에 저장됩니다.
6. 사용자 지정 프로세스 - 외부 소스에서 보안 인증을 가져옵니다.

7. 구성 파일 - `aws configure` 명령을 실행하면 `credentials` 및 `config` 파일이 업데이트됩니다. `config` 파일은 `~/.aws/config`(Linux 또는 macOS) 또는 `C:\Users\USERNAME\.aws\config`(Windows)에 저장됩니다.
8. 컨테이너 보안 인증 - IAM 역할을 각 Amazon Elastic Container Service(AmazonECS) 태스크 정의와 연결할 수 있습니다. 그러면 작업의 컨테이너에 대해 해당 역할의 임시 보안 인증을 사용할 수 있습니다. 자세한 내용은 Amazon Elastic Container Service 개발자 안내서의 IAM 작업에 대한 역할을 참조하세요.
9. Amazon EC2 인스턴스 프로필 자격 증명 - IAM 역할을 각 Amazon Elastic Compute Cloud(AmazonEC2) 인스턴스와 연결할 수 있습니다. 그러면 인스턴스에서 실행되는 코드에 대해 해당 역할의 임시 보안 인증을 사용할 수 있습니다. 자격 증명은 Amazon EC2 메타데이터 서비스를 통해 전달됩니다. 자세한 내용은 IAM Amazon 사용 설명서의 Amazon용 역할 EC2 및 IAM 사용 설명서의 인스턴스 프로파일 사용을 참조하세요. EC2

## 이 섹션의 추가 주제

- the section called “의 구성 및 보안 인증 파일 설정 AWS CLI”
- the section called “환경 변수”
- the section called “의 명령줄 옵션 AWS CLI”
- the section called “에서 명령 완료 구성 AWS CLI”
- the section called “재시도”
- the section called “에 대한 HTTP 프록시 사용 AWS CLI”

## 의 구성 및 보안 인증 파일 설정 AWS CLI

AWS CLI에서 유지되는 파일에 자주 사용되는 구성 설정과 보안 인증을 저장할 수 있습니다.

파일은 `profiles`로 나뉩니다. 기본적으로 는 라는 이름의 프로필에 있는 설정을 AWS CLI 사용합니다`default`. 대체 설정을 사용하려면 추가 프로파일을 생성해 참조할 수 있습니다.

지원되는 환경 변수 중 하나를 설정하거나 명령줄 파라미터를 사용하여 개별 설정을 재정의할 수 있습니다. 구성 설정 우선 순위에 대한 자세한 내용은 에 대한 설정 구성 AWS CLI 섹션을 참조하세요.

> **Note**
>
> 보안 인증 설정에 대한 자세한 내용은 인증 및 액세스 보안 인증 단원을 참조하세요.

주제



# 구성 및 보안 인증 파일의 형식

`config` 및 `credentials` 파일은 섹션으로 구성됩니다. 섹션에는 프로파일 및 서비스가 포함됩니다. 섹션은 이름이 지정된 설정 모음이며 다른 섹션 정의 라인을 찾을 때까지 계속됩니다. 여러 프로파일 및 섹션을 `config` 및 `credentials` 파일에 저장할 수 있습니다.

이 파일은 다음 형식을 사용하는 일반 텍스트 파일입니다.

- 섹션 이름은 괄호[]로 묶여 있습니다(예: `[default]`, `[profile` *`user1`*`]`, `[sso-session]`).
- 섹션의 모든 항목은 `setting_name=value`와 같은 일반적인 형식을 취합니다.
- 줄은 해시 문자(#)로 시작하여 주석 처리할 수 있습니다.

config 및 credentials 파일에는 다음과 같은 섹션 유형이 포함됩니다.

- 섹션 유형: profile
- 섹션 유형: services

## 섹션 유형: **`profile`**

AWS CLI 스토어

파일에 따라 프로파일 섹션 이름은 다음 형식을 사용합니다.

- Config 파일: `[default]` `[profile` *`user1`*`]`
- 보안 인증 파일: `[default]` [*`user1`*]

  `credentials` 파일에서 항목을 생성할 때에는 `profile` 단어를 사용하지 마세요.

각 프로파일은 다른 보안 인증 정보를 지정하며, 다른 AWS 리전 및 출력 형식도 지정할 수 있습니다. `config` 파일에서 프로파일 이름을 지정할 때 접두사 "`profile`“를 포함시키되`credentials` 파일에는 포함시키지 마세요.

다음 예에서는 두 개의 프로파일, 리전 및 출력이 지정된 `credentials` 및 `config` 파일을 보여 줍니다. 첫 번째 [기본값]은 지정된 프로파일이 없는 AWS CLI 명령을 실행할 때 사용됩니다. 두 번째는 `--profile user1` 파라미터로 AWS CLI 명령을 실행할 때 사용됩니다.

Short-term credentials

이 예는 AWS Identity and Access Management의 단기 보안 인증을 위한 것입니다. 자세한 내용은 the section called “단기 보안 인증” 단원을 참조하십시오.

보안 인증 파일

```
[default]
aws_access_key_id=ASIAIOSFODNN7EXAMPLE
aws_secret_access_key=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
aws_session_token
 = IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZVERYLONGSTRINGEXAMPLE

[user1]
aws_access_key_id=ASIAI44QH8DHBEXAMPLE
aws_secret_access_key=je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY
aws_session_token
 = fcZib3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZVERYLONGSTRINGEXAMPLE
```

Config 파일

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

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

IAM role

이 예제는 IAM 역할을 수임하기 위한 것입니다. IAM 역할을 사용하는 프로필은 다른 프로필에서 자격 증명을 가져온 다음 IAM 역할 권한을 적용합니다. 다음 예에서 `default`는 보안 인증의 소스 프

로필이며 user1는 동일한 보안 인증을 차용한 다음 새 역할을 맡습니다. 자세한 내용은 the section called “IAM 역할” 단원을 참조하십시오.

보안 인증 파일

credentials 파일은 소스 프로필에서 사용하는 인증에 따라 달라집니다. 다음 예제의 경우 소스 프로필은 단기 보안 인증을 사용합니다.

```
[default]
aws_access_key_id=ASIAIOSFODNN7EXAMPLE
aws_secret_access_key=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
aws_session_token
 = IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZVERYLONGSTRINGEXAMPLE
```

Config 파일

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

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

Amazon EC2 instance metadata credentials

이 예제는 호스팅 Amazon EC2 인스턴스 메타데이터에서 얻은 자격 증명에 대한 것입니다. 자세한 내용은 the section called “에서 Amazon EC2 인스턴스 메타데이터를 보안 인증으로 사용 AWS CLI” 단원을 참조하십시오.

보안 인증 파일

credentials 파일은 이 인증 방법에 사용되지 않습니다.

Config 파일

```
[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**
>
> 보안 위험을 방지하려면 특별히 제작된 소프트웨어를 개발하거나 실제 데이터로 작업할 때 IAM 사용자를 인증에 사용하지 마세요. 대신 AWS IAM Identity Center과 같은 보안 인증 공급자를 통한 페더레이션을 사용하십시오.

이 예는 AWS Identity and Access Management의 장기 보안 인증을 위한 것입니다. 자세한 내용은 the section called “IAM 사용자” 단원을 참조하십시오.

보안 인증 파일

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

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

Config 파일

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

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

자세한 내용과 추가 권한 부여 및 보안 인증 방법은 the section called “IAM 사용자”을 참조하세요.

## 섹션 유형: **services**

이 services 섹션은 AWS 서비스 요청에 대한 사용자 지정 엔드포인트를 구성하는 설정 그룹입니다. 그런 다음 프로필이 services 섹션에 연결됩니다.

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

services 섹션은<SERVICE> = 줄로 하위 섹션으로 구분되며, 여기서 <SERVICE>는 AWS 서비스 서비스 식별자 키입니다. AWS 서비스 식별자는 모든 공백을 밑줄로 바꾸고 모든 문자를 소문자로 표시serviceId하여 API 모델의 를 기반으로 합니다. services 섹션에서 사용할 모든 서비스 식별자 키 목록은 에서 엔드포인트 사용 AWS CLI을 참조하십시오. 서비스 식별자 키 뒤에는 각각 고유한 줄에 공백 두 개로 들여쓰기하여 중첩된 설정이 이어집니다.

다음 예제는 에서 Amazon DynamoDB 서비스에 대한 요청에 사용할 엔드포인트를 구성합니다.*my-services* 에서 사용되는 섹션 *dev* 프로필. 들여쓰기된 바로 다음 줄은 해당 하위 섹션에 포함되며 해당 서비스에 적용됩니다.

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

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

서비스별 엔드포인트에 대한 자세한 내용은 에서 엔드포인트 사용 AWS CLI을 참조하세요.

프로필에 수IAM임 역할 기능을 위한 source_profile 파라미터를 통해 구성된 역할 기반 자격 증명이 있는 경우 는 지정된 프로필에 대한 서비스 구성SDK만 사용합니다. 역할이 연결된 프로파일은 사용하지 않습니다. 예를 들어 다음과 같은 공유 config 파일을 사용합니다.

```
[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
```

프로필을 사용하고 코드에서 Amazon 로 전화를 B 걸면 엔드포인트EC2가 로 확인됩니다`https://profile-b-ec2-endpoint.aws`. 코드에서 다른 서비스에 요청을 하는 경우 엔드포인트 확인은 사용자 지정 로직을 따르지 않습니다. 엔드포인트는 프로파일 A에 정의된 글로벌 엔드포인트로 확인되지 않습니다. 글로벌 엔드포인트가 프로파일 B에 적용되려면 프로파일 B 내에서 직접 `endpoint_url`을 설정해야 합니다.

## 구성 설정이 저장되는 장소는 어딘가요?

는 사용자가 지정한 민감한 자격 증명 정보를 라는 `aws configure` 로컬 파일에, 홈 디렉터리`.aws`에 `credentials`라는 폴더에 AWS CLI 저장합니다. `aws configure`를 사용하여 지정하는 덜 민감한 구성 옵션은 `config`라는 로컬 파일에 저장되며, 홈 디렉터리의 `.aws` 폴더에도 저장됩니다.

> **config 파일에 보안 인증 저장**
>
> 가 파일에서 보안 인증을 읽을 수 있으므로 모든 프로필 설정을 단일 `config` 파일에 보관할 AWS CLI 수 있습니다. 동일한 이름을 공유하는 프로파일에 대한 보안 인증이 두 파일 모두에 있는 경우 보안 인증 파일의 키가 우선합니다. `credentials` 파일에 보안 인증을 보관하는 것이 좋습니다. 이러한 파일은 다양한 언어 소프트웨어 개발 키트()에서도 사용됩니다SDKs. SDKs 에 추가하여 를 사용하는 경우 보안 인증 정보를 자체 파일에 저장해야 하는지 AWS CLI 확인합니다.

홈 디렉터리 위치는 운영 체제에 따라 달라지지만 Windows에서는 `%UserProfile%` 환경 변수를, Unix 기반 시스템에서는 `$HOME` 또는 ~(물결표) 환경 변수를 사용하여 참조됩니다. `AWS_CONFIG_FILE` 및 `AWS_SHARED_CREDENTIALS_FILE` 환경 변수를 다른 로컬 경로로 설정하여 파일에 대해 기본이 아닌 위치를 지정할 수 있습니다. 세부 정보는 에 대한 환경 변수 구성 AWS CLI를 참조하세요.

AWS Identity and Access Management (IAM) 역할을 지정하는 공유 프로파일을 사용하면 가 작업을 AWS CLI 호출 AWS STS `AssumeRole`하여 임시 보안 인증을 검색합니다. 이러한 보안 인증은 (`~/.aws/cli/cache`)에 저장됩니다. 후속 AWS CLI 명령은 만료될 때까지 캐시된 임시 보안 인증 정보를 사용하며, 이 만료되면 는 AWS CLI 보안 인증 정보를 자동으로 새로 고칩니다.

## 명명된 프로파일 사용

명시적으로 정의된 프로파일이 없는 경우 해당 `default` 프로파일이 사용됩니다.

명명된 프로필을 사용하려면 `--profile` *`profile-name`* 옵션을 명령에 추가합니다. 다음 예제에서는 `user1` 프로필에 정의된 자격 증명 및 설정을 사용하여 모든 Amazon EC2 인스턴스를 나열합니다.

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

여러 명령에 대해 명명된 프로파일을 사용하려는 경우, 기본 프로파일로 `AWS_PROFILE` 환경 변수를 설정하면 모든 명령에서 매번 프로파일을 지정하는 것을 피할 수 있습니다. `--profile` 매개 변수를 사용해 설정을 변경할 수 있습니다.

Linux or macOS

```
$ export AWS_PROFILE=user1
```

Windows

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

환경 변수를 설정하는 데 `set`을 사용하면 사용되는 값이 변경되어 현재 명령 프롬프트 세션이 종료될 때까지 또는 변수를 다른 값으로 설정할 때까지 유지됩니다.

환경 변수를 설정하는 데 `setx`를 사용하면 명령 실행 후 생성한 모든 명령 셸의 값이 변경됩니다. 명령을 실행하는 시점에 이미 실행 중인 다른 명령 셸에는 영향이 미치지 않습니다. 이러한 변경 영향을 확인하려면 명령 셸을 닫고 다시 시작합니다.

환경 변수를 설정하면 기본 프로파일이 변경되어 셸 세션이 종료될 때까지 또는 변수를 다른 값으로 설정할 때까지 유지됩니다. 셸의 스타트업 스크립트에 이들 값을 배치하면 환경 변수가 향후 세션에서도 영구적으로 적용되도록 할 수 있습니다. 자세한 내용은 에 대한 환경 변수 구성 AWS CLI 단원을 참조하십시오.

## 구성 설정 지정 및 보기

명령을 사용해 구성 설정을 보고 지정하는 몇 가지 방법이 있습니다.

### `aws configure`

보안 인증 정보, 리전 및 출력 형식을 빠르게 설정하고 보려면 이 명령을 실행합니다. 다음 예제는 샘플 값을 보여줍니다.

```
$ 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

aws configure set를 사용하여 보안 인증 또는 구성 설정을 지정할 수 있습니다. --profile 설정으로 보거나 수정하려는 프로파일을 지정합니다.

예를 들어 다음 명령은 region프로파일에서 integ을 설정합니다.

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

설정을 제거하려면 텍스트 편집기에서 config 및 credentials 파일에서 설정을 수동으로 삭제합니다.

### aws configure get

aws configure get을 사용하여 설정한 보안 인증 또는 구성 설정을 검색할 수 있습니다. --profile 설정으로 보거나 수정하려는 프로파일을 지정합니다.

예를 들어 다음 명령은 region프로파일에서 integ 설정을 검색합니다.

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

출력이 비어 있으면 설정이 명시적으로 지정되지 않고 기본값을 사용합니다.

### aws configure list

구성 데이터를 나열하려면 aws configure list 명령을 사용합니다. 이 명령은 지정된 프로필에 사용되는 프로필, 액세스 키, 비밀 키 및 리전 구성 정보를 나열합니다. 각 구성 항목에 대해 값, 구성 값이 검색된 위치, 구성 변수 이름이 표시됩니다.

예를 들어 환경 변수 AWS 리전 에 를 제공하는 경우 이 명령은 구성한 리전의 이름, 이 값이 환경 변수에서 가져온 이름, 환경 변수의 이름을 보여줍니다.

역할 및 IAM Identity Center와 같은 임시 자격 증명 방법의 경우 이 명령은 일시적으로 캐시된 액세스 키를 표시하고 보안 액세스 키가 표시됩니다.

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

## 새 구성 및 보안 인증 설정 명령 예제

다음 예에서는 다양한 인증 방법에 대해 지정된 보안 인증 정보, 리전 및 출력을 사용하여 기본 프로필을 구성하는 방법을 보여 줍니다.

Short-term credentials

이 예는 AWS Identity and Access Management의 단기 보안 인증을 위한 것입니다. AWS 구성 마법사를 사용하여 초기 값을 설정한 다음 `aws configure set` 명령은 필요한 마지막 값을 할당합니다. 자세한 내용은 the section called “단기 보안 인증” 단원을 참조하십시오.

```
$ 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
 aws_session_token fcZib3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZVERYLONG
```

IAM role

이 예제는 IAM 역할을 수임하기 위한 것입니다. IAM 역할을 사용하는 프로필은 다른 프로필에서 자격 증명을 가져온 다음 IAM 역할 권한을 적용합니다. 다음 예에서 `default`는 보안 인증의 소스 프로필이며 `user1`는 동일한 보안 인증을 차용한 다음 새 역할을 맡습니다. 이 프로세스에는 마법사가 없으므로 `aws configure set` 명령을 사용하여 각 값을 설정합니다. 자세한 내용은 the section called “IAM 역할” 단원을 참조하십시오.

```
$ aws configure set role_arn arn:aws:iam::123456789012:role/defaultrole
$ aws configure set source_profile default
$ aws configure set role_session_name session_user1
$ aws configure set region us-west-2
```

```
$ aws configure set output json
```

Amazon EC2 instance metadata credentials

이 예제는 호스팅 Amazon EC2 인스턴스 메타데이터에서 얻은 자격 증명에 대한 것입니다. 이 프로세스에는 마법사가 없으므로 `aws configure set` 명령을 사용하여 각 값을 설정합니다. 자세한 내용은 the section called “에서 Amazon EC2 인스턴스 메타데이터를 보안 인증으로 사용 AWS CLI” 단원을 참조하십시오.

```
$ 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**
>
> 보안 위험을 방지하려면 특별히 제작된 소프트웨어를 개발하거나 실제 데이터로 작업할 때 IAM 사용자를 인증에 사용하지 마세요. 대신 AWS IAM Identity Center과 같은 보안 인증 공급자를 통한 페더레이션을 사용하십시오.

이 예는 AWS Identity and Access Management의 장기 보안 인증을 위한 것입니다. 자세한 내용은 the section called “IAM 사용자” 단원을 참조하십시오.

```
$ 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
```

## 지원되는 `config` 파일 설정

주제

config 파일에서는 다음 설정이 지원됩니다. 같은 이름의 환경 변수나 같은 이름의 명령줄 옵션으로 재정의되지 않는 한, 지정된(또는 기본 설정된) 프로파일에 나열된 값들이 사용됩니다. 어떤 순서 설정이 우선적으로 사용되는지에 대한 자세한 내용은 에 대한 설정 구성 AWS CLI 섹션을 참조하세요.

## Global settings(글로벌 설정)

*api_versions*

일부 AWS 서비스는 이전 버전과의 호환성을 지원하기 위해 여러 API 버전을 유지합니다. 기본적으로 AWS CLI 명령은 사용 가능한 최신 API 버전을 사용합니다. config 파일에 api_versions 설정을 포함하여 프로파일에 사용할 API 버전을 지정할 수 있습니다.

이 설정은 "중첩" 설정이며, 그 뒤에 사용할 AWS 서비스와 API 버전을 각각 식별하는 하나 이상의 들여쓰기된 줄이 표시됩니다. 사용 가능한 API 버전을 알아보려면 각 서비스의 설명서를 참조하세요.

다음 예제에서는 두 AWS 서비스에 대한 API 버전을 지정하는 방법을 보여줍니다. 이러한 API 버전은 이러한 설정이 포함된 프로파일에서 실행되는 명령에만 사용됩니다.

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

이 설정에는 동등한 수준의 환경 변수 또는 명령줄 파라미터가 없습니다.

**aws_access_key_id**

명령 요청을 인증하기 위한 자격 증명의 일부로 사용되는 AWS 액세스 키를 지정합니다. 이 키는 config 파일에 저장될 수도 있지만, credentials 파일에 저장하는 것이 좋습니다.

AWS_ACCESS_KEY_ID 환경 변수로 재정의할 수도 있습니다. 명령줄 옵션으로 액세스 키 ID를 지정할 수는 없습니다.

```
aws_access_key_id = AKIAIOSFODNN7EXAMPLE
```

**aws_secret_access_key**

명령 요청을 인증하기 위한 자격 증명의 일부로 사용되는 AWS 보안 암호 키를 지정합니다. 이 키는 config 파일에 저장될 수도 있지만, credentials 파일에 저장하는 것이 좋습니다.

AWS_SECRET_ACCESS_KEY 환경 변수로 재정의할 수도 있습니다. 명령줄 옵션으로 보안 액세스 키를 지정할 수는 없습니다.

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

**aws_session_token**

AWS 세션 토큰을 지정합니다. 세션 토큰은 수동으로 임시 보안 보안 인증을 지정하는 경우에만 필요합니다. 이 키는 config 파일에 저장될 수도 있지만, credentials 파일에 저장하는 것이 좋습니다.

AWS_SESSION_TOKEN 환경 변수로 재정의할 수도 있습니다. 명령줄 옵션으로 세션 토큰을 지정할 수는 없습니다.

```
aws_session_token = AQoEXAMPLEH4aoAH0gNCAPyJxz4BlCFFxWNE1OPTgk5TthT
+FvwqnKwRcOIfrRh3c/LTo6UDdyJwOOvEVPvLXCrrrUtdnniCEXAMPLE/
IvU1dYUg2RVAJBanLiHb4IgRmpRV3zrkuWJOgQs8IZZaIv2BXIa2R4Olgk
```

**ca_bundle**

인증서를 확인하는 데 사용되는 CA SSL 인증서 번들(.pem확장이 있는 파일)을 지정합니다.

AWS_CA_BUNDLE 환경 변수나 --ca-bundle 명령줄 옵션으로 재정의할 수도 있습니다.

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

**cli_follow_urlparam**

가 http:// 또는 로 시작하는 명령줄 파라미터의 URL 링크를 따르 AWS CLI 려는지 여부를 지정합니다https://. 활성화되면 검색된 콘텐츠가 대신 파라미터 값으로 사용됩니다URL.

- true - 기본값입니다. 지정된 경우 http:// 또는 https://로 시작하는 모든 문자열 파라미터를 가져오고 다운로드된 모든 콘텐츠가 해당 명령에 대한 파라미터 값으로 사용됩니다.
- false – 지정된 경우 AWS CLI 는 다른 문자열로 시작http://하거나 다른 문자열과 https:// 다른 파라미터 문자열 값을 처리하지 않습니다.

이 항목은 동등한 수준의 환경 변수나 명령줄 옵션을 가지고 있지 않습니다.

```
cli_follow_urlparam = false
```

**cli_history**

기본 설정은 “Disable”입니다. 이 설정은 AWS CLI에 대한 명령 기록을 활성화합니다. 이 설정을 활성화하면 는 aws 명령 기록을 AWS CLI 기록합니다.

```
cli_history = enabled
```

기록을 나열하려면 `aws history list` 명령을 사용하고 세부 정보를 보려면 `aws history show` 명령에 결과 `command_ids`를 사용할 수 있습니다. 자세한 내용은 AWS CLI 참조 가이드의 `aws history` 섹션을 참조하세요.

**cli_timestamp_format**

출력에 포함된 타임스탬프 값의 형식을 지정합니다. 다른 값 중 하나를 지정할 수 있습니다.

- iso8601 - AWS CLI 버전 2의 기본값입니다. 지정된 경우 는 ISO 8601에 따라 모든 타임스탬프를 AWS CLI 다시 포맷합니다.

  ISO 8601 형식의 타임스탬프는 다음 예제와 같습니다. 첫 번째 예제는 Z 사후 를 포함하여 Coordinated Universal Time(UTC)의 시간을 보여줍니다. 날짜와 시간은 T로 구분됩니다.

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

  다른 시간대를 지정하려면 대신 + 또는 를 Z지정-하고 원하는 시간대가 앞뒤에 있는 시간을 UTC2자리 값으로 지정합니다. 다음 예제에서는 이전 예제와 동일한 시간을 보여주지만 8시간 뒤에 있는 태평양 표준 시간으로 조정되었습니다UTC.

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

- 와이어 - AWS CLI 버전 1의 기본값입니다. 지정된 경우 는 HTTP 쿼리 응답에서 수신된 것과 정확히 동일한 모든 타임스탬프 값을 AWS CLI 표시합니다.

이 항목은 동등한 수준의 환경 변수나 명령줄 옵션을 가지고 있지 않습니다.

```
cli_timestamp_format = iso8601
```

**credential_process**

이 이 명령에 사용할 인증 자격 증명을 생성하거나 검색하기 위해 AWS CLI 실행하는 외부 명령을 지정합니다. 이 명령은 특정 형식으로 보안 인증을 반환해야 합니다. 이 설정을 사용하는 방법에 대

한 자세한 내용은 에서 외부 프로세스를 사용하여 보안 인증 정보 소싱 AWS CLI 단원을 참조하세요.

이 항목은 동등한 수준의 환경 변수나 명령줄 옵션을 가지고 있지 않습니다.

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

### credential_source

Amazon EC2 인스턴스 또는 컨테이너 내에서 가 AWS CLI role_arn 파라미터로 지정한 역할을 수임하는 데 사용할 자격 증명을 찾을 수 있는 위치를 지정하는 데 사용됩니다. source_profile과 credential_source 모두를 동일한 프로파일에서 지정할 수 없습니다.

이 파라미터는 다음 세 가지 값 중 하나를 가질 수 있습니다.

- 환경 - AWS CLI 가 환경 변수에서 소스 자격 증명을 검색하도록 지정합니다.
- Ec2InstanceMetadata - AWS CLI 가 EC2 인스턴스 프로파일에 연결된 IAM 역할을 사용하여 소스 자격 증명을 가져오도록 지정합니다.
- EcsContainer - AWS CLI 가 ECS 컨테이너에 연결된 IAM 역할을 소스 자격 증명으로 사용하도록 지정합니다.

```
credential_source = Ec2InstanceMetadata
```

### duration_seconds

역할 세션의 최대 기간(초)을 지정합니다. 이 값의 범위는 900초(15분)부터 해당 역할에 대한 최대 세션 기간 설정(최대값: 43200초)까지 가능합니다. 이는 선택적 파라미터이며 기본적으로 값이 3600초로 설정됩니다.

### endpoint_url

모든 서비스 요청에 사용되는 엔드포인트를 지정합니다. config 파일의 services 섹션에서 이 설정을 사용하면 엔드포인트가 지정된 서비스에 대해서만 사용됩니다.

다음 예제에서는 Amazon S3에 대해 글로벌 엔드포인트 http://localhost:1234 및 서비스별 엔드포인트 http://localhost:4567를 사용합니다.

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

[services s3-specific]
```

```
s3 =
  endpoint_url = http://localhost:4567
```

엔드포인트 구성 설정은 시스템 또는 사용자 환경 변수, 로컬 AWS 구성 파일 또는 명령줄에 파라미터로 명시적으로 선언되는 등 여러 위치에 있습니다. AWS CLI 엔드포인트 구성 설정은 다음 순서에 따라 우선적으로 적용됩니다.

1. `--endpoint-url` 명령줄 옵션
2. 사용 설정된 경우, 사용자 지정 엔드포인트를 무시하도록 `AWS_IGNORE_CONFIGURED_ENDPOINT_URLS` 글로벌 엔드포인트 환경 변수 또는 프로파일 설정 `ignore_configure_endpoint_urls`를 사용합니다.
3. 서비스별 환경 변수 `AWS_ENDPOINT_URL_<SERVICE>`에서 제공하는 값(예: `AWS_ENDPOINT_URL_DYNAMODB`).
4. `AWS_USE_DUALSTACK_ENDPOINT`, `AWS_USE_FIPS_ENDPOINT` 및 `AWS_ENDPOINT_URL` 환경 변수에서 제공하는 값입니다.
5. 공유 `config` 파일의 `services` 섹션 내의 `endpoint_url` 설정에서 제공하는 서비스별 엔드포인트 값.
6. 공유 `config` 파일의 `profile` 내에서 `endpoint_url` 설정에 의해 제공되는 값.
7. `use_dualstack_endpoint`, `use_fips_endpoint` 및 `endpoint_url` 설정입니다.
8. 각 URL 의 기본 엔드포인트 AWS 서비스 가 마지막으로 사용됩니다. 각 리전에서 사용할 수 있는 표준 서비스 엔드포인트 목록은 Amazon Web Services 일반 참조의 AWS 리전 및 엔드포인트를 참조하세요.

**`ignore_configure_endpoint_urls`**

활성화된 경우 는 `config` 파일에 지정된 모든 사용자 지정 엔드포인트 구성을 AWS CLI 무시합니다. 유효 값은 **`true`**및 **`false`**입니다.

```
ignore_configure_endpoint_urls = true
```

엔드포인트 구성 설정은 시스템 또는 사용자 환경 변수, 로컬 AWS 구성 파일 또는 명령줄에 파라미터로 명시적으로 선언되는 등 여러 위치에 있습니다. AWS CLI 엔드포인트 구성 설정은 다음 순서에 따라 우선적으로 적용됩니다.

1. `--endpoint-url` 명령줄 옵션
2. 사용 설정된 경우, 사용자 지정 엔드포인트를 무시하도록 `AWS_IGNORE_CONFIGURED_ENDPOINT_URLS` 글로벌 엔드포인트 환경 변수 또는 프로파일 설정 `ignore_configure_endpoint_urls`를 사용합니다.

3. 서비스별 환경 변수 AWS_ENDPOINT_URL_<SERVICE>에서 제공하는 값(예: AWS_ENDPOINT_URL_DYNAMODB).
4. AWS_USE_DUALSTACK_ENDPOINT, AWS_USE_FIPS_ENDPOINT 및 AWS_ENDPOINT_URL 환경 변수에서 제공하는 값입니다.
5. 공유 config 파일의 services 섹션 내의 endpoint_url 설정에서 제공하는 서비스별 엔드포인트 값.
6. 공유 config 파일의 profile 내에서 endpoint_url 설정에 의해 제공되는 값.
7. use_dualstack_endpoint, use_fips_endpoint 및 endpoint_url 설정입니다.
8. 각 URL 의 기본 엔드포인트 AWS 서비스 가 마지막으로 사용됩니다. 각 리전에서 사용할 수 있는 표준 서비스 엔드포인트 목록은 Amazon Web Services 일반 참조의 AWS 리전 및 엔드포인트를 참조하세요.

**external_id**

타사에서 고객 계정의 역할을 수임하는 데 사용하는 고유한 식별자를 지정합니다. 이는 ExternalId 작업의 AssumeRole 파라미터로 매핑됩니다. 이 파라미터는 역할에 대한 신뢰 정책에서 ExternalId에 값을 지정하는 경우에만 필요합니다. 자세한 내용은 IAM 사용 설명서의 타사에 AWS 리소스에 대한 액세스 권한을 부여할 때 외부 ID를 사용하는 방법을 참조하세요.

**max_attempts**

AWS CLI 재시도 핸들러가 사용하는 최대 재시도 횟수 값을 지정합니다. 여기서 초기 호출은 사용자가 제공하는 max_attempts 값으로 계산됩니다.

AWS_MAX_ATTEMPTS 환경 변수를 사용하여 이 값을 재정의할 수 있습니다.

```
max_attempts = 3
```

**mfa_serial**

역할을 수임할 때 사용할 MFA 디바이스의 식별 번호입니다. 이는 수임 중인 역할의 신뢰 정책에 MFA 인증이 필요한 조건이 포함된 경우에만 필수입니다. 값은 하드웨어 디바이스의 일련 번호(예: GAHT12345678) 또는 가상 MFA 디바이스의 Amazon 리소스 이름(ARN)(예: )일 수 있습니다arn:aws:iam::123456789012:mfa/*user*.

**output**

이 프로파일을 사용하여 요청된 명령의 기본 출력 형식을 지정합니다. 다음 값 중 하나를 지정할 수 있습니다.

- **`json`** - 출력은 JSON 문자열 형식입니다.
- **`text`** - 출력은 여러 줄의 탭으로 구분된 문자열 값으로 형식이 지정됩니다. 출력을 grep, sed 또는 awk와 같은 텍스트 프로세서로 전달하는 데 사용할 수 있습니다.
- **`table`** - 출력은 셀 테두리를 형성하기 위해 +|- 문자를 사용하여 표로 형식이 지정됩니다. 일반적으로 읽기는 쉽지만 프로그래밍 방식으로는 유용하지 않은 "인간 친화적" 형식으로 정보를 표시합니다.

AWS_DEFAULT_OUTPUT 환경 변수나 `--output` 명령줄 옵션으로 재정의할 수도 있습니다.

```
output = table
```

**`parameter_validation`**

AWS CLI 클라이언트가 파라미터를 AWS 서비스 엔드포인트로 전송하기 전에 파라미터 검증을 시도할지 여부를 지정합니다.

- true - 기본값입니다. 지정된 경우 는 명령줄 파라미터의 로컬 검증을 AWS CLI 수행합니다.
- false - 지정된 경우 AWS CLI 는 명령줄 파라미터를 AWS 서비스 엔드포인트로 보내기 전에 검증하지 않습니다.

이 항목은 동등한 수준의 환경 변수나 명령줄 옵션을 가지고 있지 않습니다.

```
parameter_validation = false
```

**`region`**

이 프로파일을 사용하여 요청 AWS 리전 된 명령에 대한 요청을 보낼 를 지정합니다.

- Amazon Web Services 일반 참조의 AWS 리전 및 엔드포인트에 나열된 대로 선택한 서비스에서 사용할 수 있는 모든 리전 코드를 지정할 수 있습니다.
- `aws_global` 를 사용하면 (AWS STS) 및 Amazon Simple Storage Service AWS Security Token Service (Amazon S3)와 같은 리전 엔드포인트 외에도 글로벌 엔드포인트를 지원하는 서비스에 대한 글로벌 엔드포인트를 지정할 수 있습니다.

AWS_DEFAULT_REGION 환경 변수 또는 `--region` 명령줄 옵션을 사용하여 이 값을 재정의할 수 있습니다.

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

### retry_mode

에서 AWS CLI 사용하는 재시도 모드를 지정합니다. 레거시(기본값), 표준 및 적응형, 이렇게 3가지 재시도 모드를 사용할 수 있습니다. 재시도에 대한 자세한 내용은 AWS CLI 에서 재시도 AWS CLI을 참조하세요.

AWS_RETRY_MODE 환경 변수를 사용하여 이 값을 재정의할 수 있습니다.

```
retry_mode = standard
```

### role_arn

AWS CLI 명령을 실행하는 데 사용할 IAM 역할의 Amazon 리소스 이름(ARN)을 지정합니다. 또한 다음 파라미터 중 하나를 지정하여 이 역할을 수임할 수 있는 권한이 있는 보안 인증을 식별해야 합니다.

- source_profile
- credential_source

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

환경 변수 AWS_ROLE_ARN이 이 설정을 재정의합니다.

웹 보안 인증 사용에 대한 자세한 내용은 the section called “웹 자격 증명을 사용한 역할 수임” 단원을 참조하세요.

### role_session_name

역할 세션에 연결할 이름을 지정합니다. 이 값은 AWS CLI 가 AssumeRole 작업을 호출할 때 RoleSessionName 파라미터에 제공되며, 는 가정된 역할 사용자 의 일부가 됩니다ARN arn:aws:sts::123456789012:assumed-role/role_name/role_session_name. 이는 선택 가능한 파라미터입니다. 이 값을 제공하지 않은 경우 세션 이름이 자동으로 생성됩니다. 이 이름은 이 세션과 연결된 항목에 대한 AWS CloudTrail 로그에 나타납니다.

```
role_session_name = maria_garcia_role
```

환경 변수 AWS_ROLE_SESSION_NAME이 이 설정을 재정의합니다.

웹 보안 인증 사용에 대한 자세한 내용은 the section called “웹 자격 증명을 사용한 역할 수임” 단원을 참조하세요.

### services

프로파일에 사용할 서비스 구성을 지정합니다.

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

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

services 섹션에 대한 자세한 내용은 the section called “services”을 참조하세요.

환경 변수 AWS_ROLE_SESSION_NAME이 이 설정을 재정의합니다.

웹 보안 인증 사용에 대한 자세한 내용은 the section called “웹 자격 증명을 사용한 역할 수임” 단원을 참조하세요.

### sdk_ua_app_id

여러 고객 애플리케이션에서 단일 를 사용하여 에 전화를 걸 AWS 계정 수 있습니다 AWS 서비스. 애플리케이션 ID는 를 사용하여 호출 집합을 수행한 소스 애플리케이션을 식별합니다 AWS 서비스. AWS SDKs 및 서비스는 이 값을 고객 커뮤니케이션에 다시 표시하는 것 외에는 이 값을 사용하거나 해석하지 않습니다. 예를 들어 이 값을 운영 이메일에 포함시켜 알림과 연결된 애플리케이션을 고유하게 식별할 수 있습니다.

애플리케이션 ID는 최대 길이가 50자인 문자열입니다. 문자, 숫자 및 특수 문자는 허용됩니다. ! $ % & * + - . , ^ _ ` | ~ 기본적으로 값이 할당되지 않습니다.

```
sdk_ua_app_id = prod1
```

AWS_SDK_UA_APP_ID 환경 변수를 사용하여 이 설정을 덮어쓸 수 있습니다. 이 값은 명령줄 파라미터로 설정할 수 없습니다.

### source_profile

AWS CLI 에서 role_arn 파라미터로 지정한 역할을 수임하는 데 사용할 수 있는 장기 보안 인증으로 명명된 프로파일을 지정합니다. source_profile과 credential_source 모두를 동일한 프로파일에서 지정할 수 없습니다.

```
source_profile = production-profile
```

*sts_regional_endpoints*

가 AWS CLI 클라이언트가 AWS Security Token Service ()와 대화하는 데 사용하는 AWS 서비스 엔드포인트를 AWS CLI 결정하는 방법을 지정합니다AWS STS. AWS CLI 버전 1의 기본값은 입니다legacy.

다음 두 값 중 하나를 지정할 수 있습니다.

- **legacy** – ap-northeast-1, , , ap-south-1, sts.amazonaws.com, , , , , , , , ap-southeast-1ap-southeast-2aws-globalca-central-1eu-central-1eu-north-1eu-west-1eu-west-2, eu-west-3, sa-east-1, , us-east-1, , , , AWS 리전에 글로벌 STS 엔드포인트 us-east-2us-west-1를 사용합니다us-west-2. 다른 모든 리전은 해당 리전 엔드포인트를 자동으로 사용합니다.
- **regional** – 는 현재 구성된 리전에 대해 AWS CLI 항상 AWS STS 엔드포인트를 사용합니다. 예를 들어 클라이언트가 를 사용하도록 구성된 경우 us-west-2에 대한 모든 호출 AWS STS 은 글로벌 엔드포인트 sts.us-west-2.amazonaws.com 대신 리전 sts.amazonaws.com 엔드포인트로 이루어집니다. 이 설정이 활성화된 상태에서 글로벌 엔드포인트에 요청을 보내려면 리전을 aws-global로 설정하면 됩니다.

AWS_STS_REGIONAL_ENDPOINTS 환경 변수를 사용하여 이 설정을 덮어쓸 수 있습니다. 이 값은 명령줄 파라미터로 설정할 수 없습니다.

*use_dualstack_endpoint*

듀얼 스택 엔드포인트를 사용하여 AWS 요청을 보낼 수 있습니다. IPv4 및 IPv6 트래픽을 모두 지원하는 듀얼 스택 엔드포인트에 대한 자세한 내용은 Amazon Simple Storage Service 사용 설명서의 Amazon S3 듀얼 스택 엔드포인트 사용을 참조하세요.  이중 스택 엔드포인트는 일부 리전에 사용할 수 있는 서비스입니다. 서비스 또는 에 대한 듀얼 스택 엔드포인트가 없는 경우 요청이 실패 AWS 리전합니다. 이 옵션은 기본적으로 비활성화되어 있습니다.

이 옵션은 use_accelerate_endpoint 설정에서 함께 사용할 수 없습니다.

엔드포인트 구성 설정은 시스템 또는 사용자 환경 변수, 로컬 AWS 구성 파일 또는 명령줄에 파라미터로 명시적으로 선언되는 등 여러 위치에 있습니다. AWS CLI 엔드포인트 구성 설정은 다음 순서에 따라 우선적으로 적용됩니다.

1. --endpoint-url 명령줄 옵션

2. 사용 설정된 경우, 사용자 지정 엔드포인트를 무시하도록 AWS_IGNORE_CONFIGURED_ENDPOINT_URLS 글로벌 엔드포인트 환경 변수 또는 프로파일 설정 ignore_configure_endpoint_urls를 사용합니다.
3. 서비스별 환경 변수 AWS_ENDPOINT_URL_<SERVICE>에서 제공하는 값(예: AWS_ENDPOINT_URL_DYNAMODB).
4. AWS_USE_DUALSTACK_ENDPOINT, AWS_USE_FIPS_ENDPOINT 및 AWS_ENDPOINT_URL 환경 변수에서 제공하는 값입니다.
5. 공유 config 파일의 services 섹션 내의 endpoint_url 설정에서 제공하는 서비스별 엔드포인트 값.
6. 공유 config 파일의 profile 내에서 endpoint_url 설정에 의해 제공되는 값.
7. use_dualstack_endpoint, use_fips_endpoint 및 endpoint_url 설정입니다.
8. 각 URL 의 기본 엔드포인트 AWS 서비스 가 마지막으로 사용됩니다. 각 리전에서 사용할 수 있는 표준 서비스 엔드포인트 목록은 Amazon Web Services 일반 참조의 AWS 리전 및 엔드포인트를 참조하세요.

*use_fips_endpoint*

일부 AWS 서비스는 일부 에서 연방 정보 처리 표준(FIPS) 140-2를 지원하는 엔드포인트를 제공합니다 AWS 리전. AWS 서비스가 를 지원하는 경우 FIPS이 설정은 에서 를 사용해야 AWS CLI 하는 FIPS 엔드포인트를 지정합니다. 표준 AWS 엔드포인트와 달리 FIPS 엔드포인트는 FIPS 140-2를 준수하는 TLS 소프트웨어 라이브러리를 사용합니다. 이러한 엔드포인트는 미국 정부와 상호 작용하는 기업에 필요할 수 있습니다.

이 설정이 활성화되었지만 의 서비스에 대한 FIPS 엔드포인트가 없는 경우 AWS 명령 AWS 리전이 실패할 수 있습니다. 이 경우 --endpoint-url 옵션을 사용하여 명령에 사용할 엔드포인트를 수동으로 지정하거나 서비스별 엔드포인트를 사용합니다.

로 FIPS 엔드포인트를 지정하는 방법에 대한 자세한 내용은 서비스별 엔드포인트 를 AWS 리전참조하세요. FIPS

엔드포인트 구성 설정은 시스템 또는 사용자 환경 변수, 로컬 AWS 구성 파일 또는 명령줄에 파라미터로 명시적으로 선언되는 등 여러 위치에 있습니다. AWS CLI 엔드포인트 구성 설정은 다음 순서에 따라 우선적으로 적용됩니다.

1. --endpoint-url 명령줄 옵션
2. 사용 설정된 경우, 사용자 지정 엔드포인트를 무시하도록 AWS_IGNORE_CONFIGURED_ENDPOINT_URLS 글로벌 엔드포인트 환경 변수 또는 프로파일 설정 ignore_configure_endpoint_urls를 사용합니다.

3. 서비스별 환경 변수 AWS_ENDPOINT_URL_<SERVICE>에서 제공하는 값(예: AWS_ENDPOINT_URL_DYNAMODB).
4. AWS_USE_DUALSTACK_ENDPOINT, AWS_USE_FIPS_ENDPOINT 및 AWS_ENDPOINT_URL 환경 변수에서 제공하는 값입니다.
5. 공유 config 파일의 services 섹션 내의 endpoint_url 설정에서 제공하는 서비스별 엔드포인트 값.
6. 공유 config 파일의 profile 내에서 endpoint_url 설정에 의해 제공되는 값.
7. use_dualstack_endpoint, use_fips_endpoint 및 endpoint_url 설정입니다.
8. 각 URL 의 기본 엔드포인트 AWS 서비스 가 마지막으로 사용됩니다. 각 리전에서 사용할 수 있는 표준 서비스 엔드포인트 목록은 Amazon Web Services 일반 참조의 AWS 리전 및 엔드포인트를 참조하세요.

*web_identity_token_file*

자격 증명 공급자가 제공하는 OAuth 2.0 액세스 토큰 또는 OpenID Connect ID 토큰이 포함된 파일의 경로를 지정합니다. AWS CLI 에서 이 파일의 내용을 로드하고 해당 파일을 WebIdentityToken 작업에 대한 AssumeRoleWithWebIdentity 인수로 전달합니다.

환경 변수 AWS_WEB_IDENTITY_TOKEN_FILE이 이 설정을 재정의합니다.

웹 보안 인증 사용에 대한 자세한 내용은 the section called “웹 자격 증명을 사용한 역할 수임” 단원을 참조하세요.

*tcp_keepalive*

AWS CLI 클라이언트가 연결 TCP 유지 패킷을 사용할지 여부를 지정합니다.

이 항목은 동등한 수준의 환경 변수나 명령줄 옵션을 가지고 있지 않습니다.

```
tcp_keepalive = false
```

## S3 사용자 지정 명령 설정

Amazon S3는 이 Amazon S3 작업을 AWS CLI 수행하는 방법을 구성하는 여러 설정을 지원합니다. 일부 설정은 s3api 및s3 네임스페이스의 모든 S3 명령에 적용됩니다. 다른 명령은 특히 공통 작업을 추상화하고 API 작업에 대한 매핑 이상을 one-to-one 수행하는 S3 '사용자 지정' 명령을 위한 것입니다. aws s3 이전 명령인 cp, sync, mv 및 rm에는 S3 이전을 제어하는 데 사용할 수 있는 추가 설정이 있습니다.

이러한 옵션은 모두 config 파일에서 s3 중첩 설정을 지정하여 구성할 수 있습니다. 각 설정은 자체의 줄에서 들여쓰기가 됩니다.

> **Note**
>
> 이러한 설정은 전적으로 선택 사항입니다. 이러한 설정을 구성하지 않고도 aws s3 이전 명령을 성공적으로 사용할 수 있어야 합니다. 이러한 설정은 성능을 조정하거나 aws s3 명령을 실행 중인 특정 환경을 설명할 수 있도록 하기 위해 제공됩니다.

이러한 설정은 모두 상위 수준 s3 파일의 config 키에서 지정됩니다(development 프로파일의 경우 다음 예제 참조).

```
[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
```

아래 설정은 s3 또는 s3api 네임스페이스의 모든 S3 명령에 적용됩니다.

**addressing_style**

사용할 주소 지정 방식을 지정합니다. 이렇게 하면 버킷 이름이 호스트 이름에 있는지 아니면 의 일부인지 제어합니다URL. 유효 값은 path, virtual 및 auto입니다. 기본값은 auto입니다.

Amazon S3 엔드포인트는 두 가지 방식으로 구성할 수 있습니다. 첫 번째는 virtual이라고 하는 방식으로 호스트 이름의 일부로 버킷 이름을 포함하고 있습니다. 예: https://*bucketname*.s3.amazonaws.com. 또는 path 스타일을 사용하면 버킷 이름을 URI의 경로인 것처럼 취급합니다. 예를 들어 입니다https://s3.amazonaws.com/*bucketname*. 의 기본값은 를 사용하는 것입니다. autoCLI는 virtual 가능한 스타일을 사용하려고 하지만 필요한 경우 path 스타일로 돌아갑니다. 예를 들어 버킷 이름이 DNS 호환되지 않는 경우 버킷 이름은 호스트 이름의 일부가 될 수 없으며 경로에 있어야 합니다. auto를 사용하면 CLI가 이 조건을 감지하고 자동으로 path 스타일로 전환합니다. 주소 지정 스타일을 로 설정한 경우 path에서 AWS CLI 구성한 AWS 리전이 버킷의 리전과 일치하는지 확인해야 합니다.

**payload_signing_enabled**

Sigv4 페이로드에 SHA256 서명할지 여부를 지정합니다. 기본적으로 를 사용할 때 스트리밍 업로드(UploadPart 및 PutObject)에는 비활성화됩니다HTTPS. 기본적으로 스트리밍 업로드(UploadPart 및 PutObject)false의 경우 로 설정되지만 이 존재하고(기본적으로 생성됨) 엔드포인트ContentMD5에서 를 사용하는 경우에만 가능합니다HTTPS.

true로 설정하면 S3 요청은 사용자에 대해 계산되고 요청 서명에 포함된 SHA256 체크섬의 형태로 추가 콘텐츠 검증을 받습니다. false로 설정되어 있는 경우에는 체크섬이 계산되지 않습니다. 이 옵션을 비활성화하는 것이 체크섬 계산에서 생성된 성능 오버헤드를 줄이는 데 유용할 수 있습니다.

**use_accelerate_endpoint**

모든 s3 및 s3api 명령에서 Amazon S3 Accelerate 엔드포인트를 사용합니다. 기본값은 false입니다. 이 옵션은 use_dualstack_endpoint 설정에서 함께 사용할 수 없습니다.

true로 설정하면 는 모든 Amazon S3 요청을 의 S3 Accelerate 엔드포인트로 AWS CLI 전달합니다s3-accelerate.amazonaws.com. 이 엔드포인트를 사용하려면 버킷에서 S3 Accelerate를 사용하도록 활성화해야 합니다. 모든 요청은 가상의 버킷 주소 지정 방식(*my-bucket*.s3-accelerate.amazonaws.com)을 사용하여 전송됩니다. 엔드포인트에서 이러한 작업을 지원하지 않기 때문에 어떤 ListBuckets, CreateBucket 및 DeleteBucket 요청도 S3 가속 엔드포인트로 전송되지 않습니다. --endpoint-url 또는 https://s3-accelerate.amazonaws.com 명령에서 http://s3-accelerate.amazonaws.com 파라미터가 s3 또는 s3api으로 설정되어 있는 경우에는 이 동작도 설정할 수 있습니다.

다음 설정은 s3 네임스페이스 명령 집합의 명령에만 적용됩니다.

**max_bandwidth**

Amazon S3의 데이터 업로드 및 다운로드에 사용할 수 있는 최대 대역폭을 지정합니다. 기본 값은 제한 없음입니다.

이 값은 S3 명령이 Amazon S3와 데이터를 주고 받는 데 사용할 수 있는 최대 대역폭을 제한합니다. 이 값은 업로드 및 다운로드에만 적용되고, 복사 또는 삭제 작업에는 적용되지 않습니다. 이 값은 초당 바이트로 표현됩니다. 이 값을 다음과 같이 형태로 지정할 수 있습니다.

- 정수. 예를 들어 1048576은 초당 1MB로 최대 대역폭 사용량을 설정합니다.
- 뒤에 속도 접미사가 붙는 정수. KB/s, MB/s 또는 GB/s를 사용하여 속도 접미사를 지정할 수 있습니다. 예, 300KB/s, 10MB/s.

일반적으로 먼저 max_concurrent_requests를 낮춰서 대역폭 사용량을 낮추려고 시도하는 것이 좋습니다. 이렇게 해도 원하는 속도로 대역폭 사용량이 적절하게 제한되지 않는 경우에는 대역폭 사용량을 추가로 제한하는 데 사용되는 max_bandwidth 설정을 사용할 수 있습니다. 이는 max_concurrent_requests가 현재 실행 중인 스레드의 수를 제어하기 때문입니다. 대신에 먼저 max_bandwidth 값을 낮추되, max_concurrent_requests 설정은 높게 놔두면 스레드가 불필요하게 대기해야 하는 결과가 발생할 수 있습니다. 이로 인해 리소스 사용량과 연결 제한 시간이 초과할 수 있습니다.

**max_concurrent_requests**

동시 요청의 최대 수를 지정합니다. 기본값은 10입니다.

aws s3 이전 명령은 멀티스레드가 됩니다. 언제든지 여러 개의 Amazon S3 요청이 실행 중일 수 있습니다. 예를 들어 명령을 사용하여 S3 버킷aws s3 cp localdir s3://bucket/ --recursive에 파일을 업로드하는 경우 는 파일 localdir/file1, localdir/file2및 를 병렬localdir/file3로 업로드할 AWS CLI 수 있습니다. max_concurrent_requests 설정은 동시에 실행 가능한 이전 작업의 최대 수를 지정합니다.

몇 가지 이유에서 이 값을 변경해야 할 수도 있습니다.

- 이 값을 줄이기 - 어떤 환경에서는 기본 설정된 10개의 동시 요청으로 인해 시스템이 압도될 수 있습니다. 이로 인해 연결 제한 시간이 발생하거나 시스템의 응답 속도가 느려질 수 있습니다. 이 값을 낮추면 S3 이전 명령에서 리소스를 덜 사용하게 됩니다. 하지만 S3 이전이 완료되는 데 더 많은 시간이 소요될 수 있다는 단점이 있습니다. 대역폭을 제한하기 위한 도구를 사용하는 경우에는 이 값을 낮추는 것이 필수로 요구될 수 있습니다.
- 이 값을 늘리기 - 어떤 경우에는 필요한 만큼 네트워크 대역폭을 사용하여 가능한 한 신속하게 Amazon S3 전송을 완료하고 싶을 수 있습니다. 이런 경우에는 기본적인 동시 요청 수로는 사용 가능한 모든 네트워크 대역폭을 활용하기에 충분하지 않을 수 있습니다. 이 값을 늘리면 Amazon S3 이전을 완료하는 데 소요되는 시간을 줄일 수 있습니다.

**max_queue_size**

작업 대기열의 최대 작업 수를 지정합니다. 기본값은 1000입니다.

는 AWS CLI 내부적으로 Amazon S3 태스크를 대기열에 넣는 모델을 사용하며, 이 작업은 숫자가 로 제한된 소비자가 실행합니다max_concurrent_requests. 태스크는 보통 단일 Amazon S3 작업에 매핑됩니다. 예를 들어 작업은 PutObjectTask, GetObjectTask 또는 UploadPartTask가 될 수 있습니다. 작업이 대기열에 추가되는 속도는 소비자가 작업을 완료하는 속도보다 훨씬 빠를 수 있습니다. 무한 증가를 피하기 위해 작업 대기열 크기가 특정 크기로 제한됩니다. 이 설정은 최대 크기의 값을 변경합니다.

일반적으로 이 설정을 변경할 필요는 없습니다. 이 설정은 AWS CLI 가 실행해야 함을 알고 있는 작업 수에도 해당합니다. 즉, 기본적으로 는 앞으로 1,000개의 작업만 볼 AWS CLI 수 있습니다. 이 값을 늘리면 대기 속도가 작업 완료 속도보다 빠르다고 가정할 때 가 필요한 총 작업 수를 더 빠르게 알 AWS CLI 수 있습니다. max_queue_size가 커질수록 메모리가 더 필요하게 된다는 단점이 있습니다.

**`multipart_chunksize`**

가 개별 파일의 멀티파트 전송에 AWS CLI 사용하는 청크 크기를 지정합니다. 기본값은 8MB이며 최소 5MB입니다.

파일 전송이 `multipart_threshold`를 초과하면 AWS CLI 는 파일을 이 크기의 청크로 분할합니다. `multipart_threshold`와 동일한 구문을 사용하여, 즉 정수 형태의 바이트 수를 사용하거나 크기와 접미사를 사용하는 방법으로 이 값을 지정할 수 있습니다.

**`multipart_threshold`**

개별 파일의 멀티파트 전송에 AWS CLI 사용하는 크기 임계값을 지정합니다. 기본값은 8MB입니다.

파일을 업로드, 다운로드 또는 복사할 때 파일이 이 크기를 초과하면 Amazon S3 명령이 멀티파트 작업으로 전환됩니다. 이 값을 두 가지 방법으로 지정할 수 있습니다.

- 먼저 바이트 단위의 파일 크기입니다. 예: `1048576`.
- 두 번째는 크기 접미사가 포함된 파일 크기입니다. `KB`, `MB`, `GB` 또는 `TB`를 사용할 수 있습니다. 예: `10MB`, `1GB`.

> **Note**
>
> S3은 멀티파트 작업에 사용할 수 있는 유효 값에 제약을 둘 수 있습니다. 자세한 내용은 Amazon Simple Storage Service 사용 설명서에서 S3 멀티파트 업로드 설명서를 참조하세요.

# 에 대한 환경 변수 구성 AWS CLI

환경 변수는 구성 옵션과 보안 인증을 지정하는 또 다른 방법을 제공하며 스크립팅에 유용할 수 있습니다.

옵션의 우선 순위

- 이 주제에 설명된 환경 변수 중 하나를 사용하여 옵션을 지정할 경우, 구성 파일의 프로파일에서 로드된 값을 재정의합니다.

- AWS CLI 명령줄의 파라미터를 사용하여 옵션을 지정하는 경우 해당 환경 변수 또는 구성 파일의 프로파일에서 모든 값을 재정의합니다.

우선 순위 및 에서 사용할 보안 인증 정보를 AWS CLI 결정하는 방법에 대한 자세한 내용은 섹션을 참조하세요에 대한 설정 구성 AWS CLI.

주제



## 환경 변수를 설정하는 방법

다음은 기본 사용자에 대한 환경 변수를 구성할 수 있는 방법을 보여주는 예입니다.

Linux or macOS

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

환경 변수를 설정하면 사용되는 값이 변경되어 쉘 세션이 종료될 때까지 또는 변수를 다른 값으로 설정할 때까지 유지됩니다. 쉘의 스타트업 스크립트에서 변수를 설정하면 해당 변수가 향후 세션에서도 영구적으로 적용되도록 할 수 있습니다.

Windows Command Prompt

모든 세션에 대해 설정하려면

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

환경 변수를 설정하는 데 `setx`를 사용하면 현재 명령 프롬프트 세션과 명령 실행 후 생성한 모든 명령 프롬프트 세션에서 사용되는 값이 변경됩니다. 명령을 실행하는 시점에 이미 실행 중인 다른 명령 쉘에는 영향을 주지 않습니다. 설정을 로드하려면 터미널을 다시 시작해야 할 수 있습니다.

현재 세션에만 설정하려면

환경 변수를 설정하는 데 `set`을 사용하면 사용되는 값이 변경되어 현재 명령 프롬프트 세션이 종료될 때까지 또는 변수를 다른 값으로 설정할 때까지 유지됩니다.

```
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"
```

이전 예제와 같이 PowerShell 프롬프트에서 환경 변수를 설정하면 현재 세션 기간 동안만 값이 저장됩니다. 모든 PowerShell 및 명령 프롬프트 세션에서 환경 변수 설정을 지속하려면 제어판의 시스템 애플리케이션을 사용하여 저장합니다. 또는 PowerShell 프로파일에 변수를 추가하여 향후 모든 PowerShell 세션에 대한 변수를 설정할 수 있습니다. 환경 변수를 저장하거나 세션 간에 유지하는 방법에 대한 자세한 내용은 PowerShell 설명서를 참조하세요.

## AWS CLI 지원되는 환경 변수

는 다음 환경 변수를 AWS CLI 지원합니다.

**AWS_ACCESS_KEY_ID**

IAM 계정과 연결된 AWS 액세스 키를 지정합니다.

정의된 경우 이 환경 변수는 aws_access_key_id 프로파일 설정 값을 재정의합니다. 명령줄 옵션으로 액세스 키 ID를 지정할 수는 없습니다.

**AWS_CA_BUNDLE**

인증서 검증에 사용할 HTTPS 인증서 번들의 경로를 지정합니다.

정의된 경우 이 환경 변수는 ca_bundle 프로파일 설정 값을 재정의합니다. --ca-bundle 명령줄 파라미터를 사용하여 이 환경 변수를 재정의할 수 있습니다.

**AWS_CLI_S3_MV_VALIDATE_SAME_S3_PATHS**

사용자 지정 s3 mv 명령을 사용할 때 소스 버킷과 대상 버킷이 동일한 경우 소스 파일 또는 객체를 자체로 이동할 수 있으므로 소스 파일 또는 객체가 실수로 삭제될 수 있습니다. AWS_CLI_S3_MV_VALIDATE_SAME_S3_PATHS 환경 변수 및 --validate-same-s3-paths 옵션은 Amazon S3 소스 또는 대상 에서 액세스 포인트 ARNs 또는 액세스 포인트 별칭을 검증할지 여부를 지정합니다URIs.

> **Note**
>
> 에 대한 경로 검증에는 추가 API 호출이 `s3 mv` 필요합니다.

**`AWS_CONFIG_FILE`**

가 구성 프로파일을 저장하는 데 AWS CLI 사용하는 파일의 위치를 지정합니다. 기본 경로는 `~/.aws/config`입니다.

명명된 프로파일 설정에서 또는 명령줄 파라미터를 사용하여 이 값을 지정할 수 없습니다.

**`AWS_DATA_PATH`**

AWS CLI 데이터를 로드할 `~/.aws/models` 때 의 기본 제공 검색 경로 외부에서 확인할 추가 디렉터리 목록입니다. 이 환경 변수를 설정하면 기본 제공 검색 경로로 넘어가기 전에 먼저 확인할 추가 디렉터리가 표시됩니다. 여러 항목은 `os.pathsep` 문자로 구분해야 합니다. 이 문자는 Linux 또는 macOS의 경우 `:`이고, Windows의 경우 `;`입니다.

**`AWS_DEFAULT_OUTPUT`**

사용할 출력 형식을 지정합니다.

정의된 경우 이 환경 변수는 `output` 프로파일 설정 값을 재정의합니다. `--output` 명령줄 파라미터를 사용하여 이 환경 변수를 재정의할 수 있습니다.

**`AWS_DEFAULT_REGION`**

는 기본적으로 요청을 보내려는 서버가 있는 AWS 리전을 `Default region name` 식별합니다. 이 리전은 일반적으로 가장 가까운 리전이지만 어떤 리전이든 될 수 있습니다. 예를 들어 미국 서부(오레곤)를 사용하려면 `us-west-2`를 입력하면 됩니다. 개별 명령으로 달리 지정하지 않는 한 이후의 모든 요청이 전송되는 리전입니다.

> **Note**
>
> 를 사용할 때 AWS CLI명시적으로 또는 기본 AWS 리전을 설정하여 리전을 지정해야 합니다. 사용 가능한 리전 목록은 리전 및 엔드포인트를 참조하세요. 에서 사용하는 리전 지정자는 및 서비스 엔드포인트에 AWS Management Console URLs 표시되는 것과 동일한 이름 AWS CLI 입니다.

정의된 경우 이 환경 변수는 region 프로파일 설정 값을 재정의합니다. --region 명령줄 파라미터수 있습니다.

**AWS_EC2_METADATA_DISABLED**

Amazon EC2 인스턴스 메타데이터 서비스() 사용을 비활성화합니다IMDS.

true로 설정하면 사용자 자격 증명 또는 구성(예: 리전)이 에서 요청되지 않습니다IMDS.

**AWS_ENDPOINT_URL**

모든 서비스 요청에 사용되는 엔드포인트를 지정합니다.

엔드포인트 구성 설정은 시스템 또는 사용자 환경 변수, 로컬 AWS 구성 파일 또는 명령줄에 파라미터로 명시적으로 선언되는 등 여러 위치에 있습니다. AWS CLI 엔드포인트 구성 설정은 다음 순서에 따라 우선적으로 적용됩니다.

1. --endpoint-url 명령줄 옵션
2. 사용 설정된 경우, 사용자 지정 엔드포인트를 무시하도록 AWS_IGNORE_CONFIGURED_ENDPOINT_URLS 글로벌 엔드포인트 환경 변수 또는 프로파일 설정 ignore_configure_endpoint_urls를 사용합니다.
3. 서비스별 환경 변수 AWS_ENDPOINT_URL_<SERVICE>에서 제공하는 값(예: AWS_ENDPOINT_URL_DYNAMODB).
4. AWS_USE_DUALSTACK_ENDPOINT, AWS_USE_FIPS_ENDPOINT 및 AWS_ENDPOINT_URL 환경 변수에서 제공하는 값입니다.
5. 공유 config 파일의 services 섹션 내의 endpoint_url 설정에서 제공하는 서비스별 엔드포인트 값.
6. 공유 config 파일의 profile 내에서 endpoint_url 설정에 의해 제공되는 값.
7. use_dualstack_endpoint, use_fips_endpoint 및 endpoint_url 설정입니다.
8. 각 URL 의 기본 엔드포인트 AWS 서비스 가 마지막으로 사용됩니다. 각 리전에서 사용할 수 있는 표준 서비스 엔드포인트 목록은 Amazon Web Services 일반 참조의 AWS 리전 및 엔드포인트를 참조하세요.

**AWS_ENDPOINT_URL_<SERVICE>**

특정 서비스에 사용되는 사용자 지정 엔드포인트를 지정합니다. 여기서 <SERVICE> 는 AWS 서비스 식별자로 대체됩니다. 예를 들어 Amazon DynamoDB 의 는 serviceId입니다DynamoDB. 이 서비스의 경우 엔드포인트 URL 환경 변수는 입니다AWS_ENDPOINT_URL_DYNAMODB.

모든 서비스별 환경 변수 목록은 서비스별 식별자 목록을 참조하십시오.

엔드포인트 구성 설정은 시스템 또는 사용자 환경 변수, 로컬 AWS 구성 파일 또는 명령줄에 파라미터로 명시적으로 선언되는 등 여러 위치에 있습니다. AWS CLI 엔드포인트 구성 설정은 다음 순서에 따라 우선적으로 적용됩니다.

1. `--endpoint-url` 명령줄 옵션
2. 사용 설정된 경우, 사용자 지정 엔드포인트를 무시하도록 `AWS_IGNORE_CONFIGURED_ENDPOINT_URLS` 글로벌 엔드포인트 환경 변수 또는 프로파일 설정 `ignore_configure_endpoint_urls`를 사용합니다.
3. 서비스별 환경 변수 `AWS_ENDPOINT_URL_<SERVICE>`에서 제공하는 값(예: AWS_ENDPOINT_URL_DYNAMODB).
4. `AWS_USE_DUALSTACK_ENDPOINT`, `AWS_USE_FIPS_ENDPOINT` 및 `AWS_ENDPOINT_URL` 환경 변수에서 제공하는 값입니다.
5. 공유 `config` 파일의 `services` 섹션 내의 `endpoint_url` 설정에서 제공하는 서비스별 엔드포인트 값.
6. 공유 `config` 파일의 `profile` 내에서 `endpoint_url` 설정에 의해 제공되는 값.
7. `use_dualstack_endpoint`, `use_fips_endpoint` 및 `endpoint_url` 설정입니다.
8. 각 URL 의 기본 엔드포인트 AWS 서비스 가 마지막으로 사용됩니다. 각 리전에서 사용할 수 있는 표준 서비스 엔드포인트 목록은 Amazon Web Services 일반 참조의 AWS 리전 및 엔드포인트를 참조하세요.

**AWS_IGNORE_CONFIGURED_ENDPOINT_URLS**

활성화된 경우 는 모든 사용자 지정 엔드포인트 구성을 AWS CLI 무시합니다. 유효 값은 **true**및 **false**입니다.

엔드포인트 구성 설정은 시스템 또는 사용자 환경 변수, 로컬 AWS 구성 파일 또는 명령줄에 파라미터로 명시적으로 선언되는 등 여러 위치에 있습니다. AWS CLI 엔드포인트 구성 설정은 다음 순서에 따라 우선적으로 적용됩니다.

1. `--endpoint-url` 명령줄 옵션
2. 사용 설정된 경우, 사용자 지정 엔드포인트를 무시하도록 `AWS_IGNORE_CONFIGURED_ENDPOINT_URLS` 글로벌 엔드포인트 환경 변수 또는 프로파일 설정 `ignore_configure_endpoint_urls`를 사용합니다.
3. 서비스별 환경 변수 `AWS_ENDPOINT_URL_<SERVICE>`에서 제공하는 값(예: AWS_ENDPOINT_URL_DYNAMODB).
4. `AWS_USE_DUALSTACK_ENDPOINT`, `AWS_USE_FIPS_ENDPOINT` 및 `AWS_ENDPOINT_URL` 환경 변수에서 제공하는 값입니다.

5. 공유 `config` 파일의 `services` 섹션 내의 `endpoint_url` 설정에서 제공하는 서비스별 엔드포인트 값.
6. 공유 `config` 파일의 `profile` 내에서 `endpoint_url` 설정에 의해 제공되는 값.
7. `use_dualstack_endpoint`, `use_fips_endpoint` 및 `endpoint_url` 설정입니다.
8. 각 URL 의 기본 엔드포인트 AWS 서비스 가 마지막으로 사용됩니다. 각 리전에서 사용할 수 있는 표준 서비스 엔드포인트 목록은 Amazon Web Services 일반 참조의 AWS 리전 및 엔드포인트를 참조하세요.

**`AWS_MAX_ATTEMPTS`**

AWS CLI 재시도 핸들러가 사용하는 최대 재시도 횟수 값을 지정합니다. 여기서 초기 호출은 사용자가 제공하는 값으로 계산됩니다. 재시도에 대한 자세한 내용은 AWS CLI 에서 재시도 AWS CLI을 참조하세요.

정의된 경우 이 환경 변수는 프로파일 설정 `max_attempts` 값을 재정의합니다.

**`AWS_METADATA_SERVICE_NUM_ATTEMPTS`**

IAM 역할로 구성된 Amazon EC2 인스턴스에서 자격 증명을 검색하려고 하면 는 인스턴스 메타데이터 서비스에서 자격 증명을 한 번 검색한 후 중지하려고 AWS CLI 시도합니다. Amazon EC2 인스턴스에서 명령이 실행될 것임을 알고 있는 경우 포기하기 전에 이 값을 늘려 여러 번 AWS CLI 재시도를 할 수 있습니다.

**`AWS_METADATA_SERVICE_TIMEOUT`**

인스턴스 메타데이터 서비스에 대한 연결 시간이 초과되기까지 경과하는 시간(초)입니다. IAM 역할로 구성된 Amazon EC2 인스턴스에서 보안 인증 정보를 검색하려고 할 때 인스턴스 메타데이터 서비스에 대한 연결은 기본적으로 1초 후에 시간 초과됩니다. IAM 역할이 구성된 Amazon EC2 인스턴스에서 실행 중인 경우 필요한 경우 이 값을 늘릴 수 있습니다.

**`AWS_PROFILE`**

사용할 자격 증명 및 옵션이 있는 AWS CLI 프로필의 이름을 지정합니다. 이 이름은 `credentials` 또는 `config` 파일에 저장된 프로필 이름이거나 기본 프로필을 사용할 값 `default`일 수 있습니다.

정의된 경우 이 환경 변수는 구성 파일에서 `[default]`라는 프로파일을 사용할 때의 동작을 재정의합니다. `--profile` 명령줄 파라미터를 사용하여 이 환경 변수를 재정의할 수 있습니다.

**`AWS_RETRY_MODE`**

에서 AWS CLI 사용하는 재시도 모드를 지정합니다. 레거시(기본값), 표준 및 적응형, 이렇게 3가지 재시도 모드를 사용할 수 있습니다. 재시도에 대한 자세한 내용은 AWS CLI 에서 재시도 AWS CLI을 참조하세요.

정의된 경우 이 환경 변수는 프로파일 설정 `retry_mode` 값을 재정의합니다.

**`AWS_ROLE_ARN`**

AWS CLI 명령을 실행하는 데 사용할 웹 자격 증명 공급자가 있는 IAM 역할의 Amazon 리소스 이름(ARN)을 지정합니다.

`AWS_WEB_IDENTITY_TOKEN_FILE` 및 `AWS_ROLE_SESSION_NAME` 환경 변수와 함께 사용됩니다.

정의된 경우 이 환경 변수는 프로파일 설정 `role_arn` 값을 재정의합니다. 명령줄 파라미터로 역할 세션 이름을 지정할 수 없습니다.

> **Note**
>
> 이러한 환경 변수는 웹 보안 인증 공급자를 사용한 역할 수임에만 적용되며 일반적인 역할 수임 공급자 구성에는 적용되지 않습니다.

웹 보안 인증 사용에 대한 자세한 내용은 the section called “웹 자격 증명을 사용한 역할 수임” 단원을 참조하세요.

**`AWS_ROLE_SESSION_NAME`**

역할 세션에 연결할 이름을 지정합니다. 이 값은 가 `AssumeRole` 작업을 AWS CLI 호출할 때 `RoleSessionName` 파라미터에 제공되며, 는 가정된 역할 사용자 의 일부가 됩니다ARN `arn:aws:sts::123456789012:assumed-role/role_name/role_session_name`. 이는 선택 가능한 파라미터입니다. 이 값을 제공하지 않은 경우 세션 이름이 자동으로 생성됩니다. 이 이름은 이 세션과 연결된 항목의 AWS CloudTrail 로그에 표시됩니다.

정의된 경우 이 환경 변수는 프로파일 설정 `role_session_name` 값을 재정의합니다.

`AWS_ROLE_ARN` 및 `AWS_WEB_IDENTITY_TOKEN_FILE` 환경 변수와 함께 사용됩니다.

웹 보안 인증 사용에 대한 자세한 내용은 the section called “웹 자격 증명을 사용한 역할 수임” 단원을 참조하세요.

> **Note**
>
> 이러한 환경 변수는 웹 보안 인증 공급자를 사용한 역할 수임에만 적용되며 일반적인 역할 수임 공급자 구성에는 적용되지 않습니다.

**AWS_SDK_UA_APP_ID**

여러 고객 애플리케이션에서 단일 를 사용하여 에 전화를 걸 AWS 계정 수 있습니다 AWS 서비스. 애플리케이션 ID는 를 사용하여 호출 집합을 수행한 소스 애플리케이션을 식별합니다 AWS 서비스. AWS SDKs 및 서비스는 이 값을 고객 커뮤니케이션에 다시 표시하는 것 외에는 이 값을 사용하거나 해석하지 않습니다. 예를 들어 이 값을 운영 이메일에 포함시켜 알림과 연결된 애플리케이션을 고유하게 식별할 수 있습니다.

기본적으로 값은 없습니다.

애플리케이션 ID는 최대 길이가 50자인 문자열입니다. 문자, 숫자 및 다음 특수 문자는 허용됩니다.

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

정의된 경우 이 환경 변수는 sdk_ua_app_id 프로파일 설정 값을 재정의합니다. 애플리케이션 ID를 명령줄 옵션으로 지정할 수 없습니다.

**AWS_SECRET_ACCESS_KEY**

액세스 키와 연결된 보안 키를 지정합니다. 이는 액세스 키에 대한 기본적인 "암호"입니다.

정의된 경우 이 환경 변수는 aws_secret_access_key 프로파일 설정 값을 재정의합니다. 명령줄 옵션으로 보안 액세스 키 ID를 지정할 수는 없습니다.

**AWS_SESSION_TOKEN**

AWS STS 작업에서 직접 검색한 임시 보안 보안 인증을 사용하는 경우 필요한 세션 토큰 값을 지정합니다. 자세한 내용은 AWS CLI 명령 참조에서 assume-role 명령의 출력 섹션을 참조하세요.

정의된 경우 이 환경 변수는 aws_session_token 프로파일 설정 값을 재정의합니다.

**AWS_SHARED_CREDENTIALS_FILE**

가 액세스 키를 저장하는 데 AWS CLI 사용하는 파일의 위치를 지정합니다. 기본 경로는 ~/.aws/credentials입니다.

명명된 프로파일 설정에서 또는 명령줄 파라미터를 사용하여 이 값을 지정할 수 없습니다.

**AWS_STS_REGIONAL_ENDPOINTS**

가 AWS CLI 클라이언트가 AWS Security Token Service ()와 대화하는 데 사용하는 AWS 서비스 엔드포인트를 AWS CLI 결정하는 방법을 지정합니다AWS STS. AWS CLI 버전 1의 기본값은 legacy입니다.

다음 두 값 중 하나를 지정할 수 있습니다.

- **legacy** - ap-northeast-1, , ap-south-1, sts.amazonaws.com, , , , , , , , , ap-southeast-1ap-southeast-2aws-globalca-central-1eu-central-1eu-north-1eu-west-1eu-west-2, eu-west-3, sa-east-1, us-east-1, us-east-2, , , AWS 리전에 글로벌 STS 엔드포인트 us-west-1를 사용합니다us-west-2. 다른 모든 리전은 해당 리전 엔드포인트를 자동으로 사용합니다.
- **regional** – 는 현재 구성된 리전에 대해 AWS CLI 항상 AWS STS 엔드포인트를 사용합니다. 예를 들어 클라이언트가 를 사용하도록 구성된 경우 us-west-2에 대한 모든 호출 AWS STS 은 글로벌 엔드포인트 sts.us-west-2.amazonaws.com 대신 리전 sts.amazonaws.com 엔드포인트로 이루어집니다. 이 설정이 활성화된 상태에서 글로벌 엔드포인트에 요청을 보내려면 리전을 aws-global로 설정하면 됩니다.

**AWS_USE_DUALSTACK_ENDPOINT**

듀얼 스택 엔드포인트를 사용하여 AWS 요청을 보낼 수 있습니다. IPv4 및 IPv6 트래픽을 모두 지원하는 듀얼 스택 엔드포인트에 대한 자세한 내용은 Amazon Simple Storage Service 사용 설명서의 Amazon S3 듀얼 스택 엔드포인트 사용을 참조하세요. 이중 스택 엔드포인트는 일부 리전에 사용할 수 있는 서비스입니다. 서비스 또는 에 대한 듀얼 스택 엔드포인트가 없는 경우 요청이 실패 AWS 리전합니다. 이 옵션은 기본적으로 비활성화되어 있습니다.

엔드포인트 구성 설정은 시스템 또는 사용자 환경 변수, 로컬 AWS 구성 파일 또는 명령줄에 파라미터로 명시적으로 선언되는 등 여러 위치에 있습니다. AWS CLI 엔드포인트 구성 설정은 다음 순서에 따라 우선적으로 적용됩니다.

1. --endpoint-url 명령줄 옵션
2. 사용 설정된 경우, 사용자 지정 엔드포인트를 무시하도록 AWS_IGNORE_CONFIGURED_ENDPOINT_URLS 글로벌 엔드포인트 환경 변수 또는 프로파일 설정 ignore_configure_endpoint_urls를 사용합니다.
3. 서비스별 환경 변수 AWS_ENDPOINT_URL_<SERVICE>에서 제공하는 값(예: AWS_ENDPOINT_URL_DYNAMODB).
4. AWS_USE_DUALSTACK_ENDPOINT, AWS_USE_FIPS_ENDPOINT 및 AWS_ENDPOINT_URL 환경 변수에서 제공하는 값입니다.
5. 공유 config 파일의 services 섹션 내의 endpoint_url 설정에서 제공하는 서비스별 엔드포인트 값.
6. 공유 config 파일의 profile 내에서 endpoint_url 설정에 의해 제공되는 값.
7. use_dualstack_endpoint, use_fips_endpoint 및 endpoint_url 설정입니다.

8. 각 URL 의 기본 엔드포인트 AWS 서비스 가 마지막으로 사용됩니다. 각 리전에서 사용할 수 있는 표준 서비스 엔드포인트 목록은 Amazon Web Services 일반 참조의 AWS 리전 및 엔드포인트를 참조하세요.

**AWS_USE_FIPS_ENDPOINT**

일부 AWS 서비스는 일부 에서 연방 정보 처리 표준(FIPS) 140-2를 지원하는 엔드포인트를 제공합니다 AWS 리전. AWS 서비스가 를 지원하는 경우 FIPS이 설정은 AWS CLI 에서 를 사용해야 하는 FIPS 엔드포인트를 지정합니다. 표준 AWS 엔드포인트와 달리 FIPS 엔드포인트는 FIPS 140-2를 준수하는 TLS 소프트웨어 라이브러리를 사용합니다. 이러한 엔드포인트는 미국 정부와 상호 작용하는 기업에 필요할 수 있습니다.

이 설정이 활성화되었지만 의 서비스에 대한 FIPS 엔드포인트가 없는 경우 AWS 명령 AWS 리전이 실패할 수 있습니다. 이 경우 `--endpoint-url` 옵션을 사용하여 명령에 사용할 엔드포인트를 수동으로 지정하거나 서비스별 엔드포인트를 사용합니다.

로 FIPS 엔드포인트를 지정하는 방법에 대한 자세한 내용은 서비스별 엔드포인트 를 AWS 리전참조하세요. FIPS

엔드포인트 구성 설정은 시스템 또는 사용자 환경 변수, 로컬 AWS 구성 파일 또는 명령줄에 파라미터로 명시적으로 선언되는 등 여러 위치에 있습니다. AWS CLI 엔드포인트 구성 설정은 다음 순서에 따라 우선적으로 적용됩니다.

1. `--endpoint-url` 명령줄 옵션
2. 사용 설정된 경우, 사용자 지정 엔드포인트를 무시하도록 `AWS_IGNORE_CONFIGURED_ENDPOINT_URLS` 글로벌 엔드포인트 환경 변수 또는 프로파일 설정 `ignore_configure_endpoint_urls`를 사용합니다.
3. 서비스별 환경 변수 `AWS_ENDPOINT_URL_<SERVICE>`에서 제공하는 값(예: AWS_ENDPOINT_URL_DYNAMODB).
4. `AWS_USE_DUALSTACK_ENDPOINT`, `AWS_USE_FIPS_ENDPOINT` 및 `AWS_ENDPOINT_URL` 환경 변수에서 제공하는 값입니다.
5. 공유 `config` 파일의 `services` 섹션 내의 `endpoint_url` 설정에서 제공하는 서비스별 엔드포인트 값.
6. 공유 `config` 파일의 `profile` 내에서 `endpoint_url` 설정에 의해 제공되는 값.
7. `use_dualstack_endpoint`, `use_fips_endpoint` 및 `endpoint_url` 설정입니다.
8. 각 URL 의 기본 엔드포인트 AWS 서비스 가 마지막으로 사용됩니다. 각 리전에서 사용할 수 있는 표준 서비스 엔드포인트 목록은 Amazon Web Services 일반 참조의 AWS 리전 및 엔드포인트를 참조하세요.

**AWS_WEB_IDENTITY_TOKEN_FILE**

자격 증명 공급자가 제공하는 OAuth 2.0 액세스 토큰 또는 OpenID Connect ID 토큰이 포함된 파일의 경로를 지정합니다. AWS CLI 에서 이 파일의 내용을 로드하고 해당 파일을 `WebIdentityToken` 작업에 대한 `AssumeRoleWithWebIdentity` 인수로 전달합니다.

`AWS_ROLE_ARN` 및 `AWS_ROLE_SESSION_NAME` 환경 변수와 함께 사용됩니다.

정의된 경우 이 환경 변수는 `web_identity_token_file` 프로파일 설정 값을 재정의합니다.

웹 보안 인증 사용에 대한 자세한 내용은 the section called “웹 자격 증명을 사용한 역할 수임” 단원을 참조하세요.

> **Note**
>
> 이러한 환경 변수는 웹 자격 증명 공급자를 사용한 역할 수임에만 적용되며 일반적인 역할 수임 공급자 구성에는 적용되지 않습니다.

# 의 명령줄 옵션 AWS CLI

에서 AWS CLI명령줄 옵션은 기본 구성 설정, 해당 프로필 설정 또는 해당 단일 명령에 대한 환경 변수 설정을 재정의하는 데 사용할 수 있는 전역 파라미터입니다. 명령줄 옵션을 통해 사용할 프로필을 지정할 수 있지만, 해당 옵션으로 보안 인증을 직접 지정할 수 없습니다.

주제



## 명령줄 옵션 사용 방법

대부분의 명령줄 옵션은 다음 예에 나온 프로파일 이름 `profile1`과 같은 단순한 문자열입니다.

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

인수를 가져오는 각 옵션에서는 공백이나 등호(=)를 사용하여 인수를 옵션 이름과 구분해야 합니다. 인수 값이 공백이 포함된 문자열인 경우 해당 인수의 앞뒤에 따옴표를 사용해야 합니다. 파라미터의 인수 유형 및 형식에 대한 자세한 내용은 에서 파라미터 값 지정 AWS CLI 단원을 참조하세요.

## AWS CLI 지원되는 전역 명령줄 옵션

에서 다음 명령줄 옵션을 사용하여 기본 구성 설정, 해당 프로파일 설정 또는 해당 단일 명령에 대한 환경 변수 설정을 재정의할 AWS CLI 수 있습니다.

--ca-번들 ***<string>***

인증서를 확인할 때 사용할 인증 기관(CA) SSL 인증서 번들을 지정합니다.

정의된 경우 이 옵션은 프로파일 설정 `ca_bundle`의 값 및 `AWS_CA_BUNDLE` 환경 변수를 재정의합니다.

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

최대 소켓 연결 시간을 초 단위로 지정합니다. 이 값이 0으로 설정되어 있으면 소켓 연결이 무한 대기 상태(차단 상태)가 되고 제한 시간이 적용되지 않습니다.

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

최대 소켓 읽기 시간을 초 단위로 지정합니다. 이 값이 0으로 설정되어 있으면 소켓 읽기가 무한 대기 상태(차단 상태)가 되고 제한 시간이 적용되지 않습니다.

--색상 ***<string>***

색상 출력에 대한 지원 여부를 지정합니다. 유효 값은 on, `off` 및 `auto`입니다. 기본 값은 `auto`입니다.

--디버그

디버그 로깅을 활성화하는 부울 스위치입니다. AWS CLI 기본적으로 는 명령 출력의 명령 결과에 관한 성공 또는 실패에 대한 정리 정보를 제공합니다. `--debug` 옵션은 전체 Python 로그를 제공합니다. 여기에는 해당 명령의 작동에 대한 추가적인 `stderr` 진단 정보가 포함되어 있는데, 이는 명령이 예기치 않은 결과를 제공하는 이유를 해결할 때 유용할 수 있습니다. 디버그 로그를 쉽게 보려면 정보를 쉽게 검색할 수 있도록 로그를 파일로 보내는 것이 좋습니다. 이를 위해 다음 중 하나를 사용할 수 있습니다.

`stderr` 진단 정보만 보내려면 `2> debug.txt`를 추가합니다. 여기서 `debug.txt`는 디버그 파일에 사용할 이름입니다.

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

출력과 stderr 진단 정보둘 다 보내려면 &> debug.txt를 추가합니다. 여기서 debug.txt는 디버그 파일에 사용할 이름입니다.

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

--엔드포인트-url *<string>*

요청을 URL 보낼 를 지정합니다. 대부분의 명령에서 는 선택한 서비스와 지정된 AWS 리전을 URL 기반으로 를 AWS CLI 자동으로 결정합니다. 그러나 일부 명령은 계정별 를 지정해야 합니다URL. 프라이빗 내에서 엔드포인트를 직접 호스팅VPC하도록 일부 AWS 서비스를 구성할 수도 있습니다. 그러면 엔드포인트를 지정해야 할 수 있습니다.

다음 명령 예제에서는 사용자 지정 Amazon S3 엔드포인트 를 사용합니다URL.

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

엔드포인트 구성 설정은 시스템 또는 사용자 환경 변수, 로컬 AWS 구성 파일 또는 명령줄에 파라미터로 명시적으로 선언되는 등 여러 위치에 있습니다. AWS CLI 엔드포인트 구성 설정은 다음 순서에 따라 우선적으로 적용됩니다.

1. --endpoint-url 명령줄 옵션
2. 사용 설정된 경우, 사용자 지정 엔드포인트를 무시하도록 AWS_IGNORE_CONFIGURED_ENDPOINT_URLS 글로벌 엔드포인트 환경 변수 또는 프로파일 설정 ignore_configure_endpoint_urls를 사용합니다.
3. 서비스별 환경 변수 AWS_ENDPOINT_URL_<SERVICE>에서 제공하는 값(예: AWS_ENDPOINT_URL_DYNAMODB).
4. AWS_USE_DUALSTACK_ENDPOINT, AWS_USE_FIPS_ENDPOINT 및 AWS_ENDPOINT_URL 환경 변수에서 제공하는 값입니다.
5. 공유 config 파일의 services 섹션 내의 endpoint_url 설정에서 제공하는 서비스별 엔드포인트 값.
6. 공유 config 파일의 profile 내에서 endpoint_url 설정에 의해 제공되는 값.
7. use_dualstack_endpoint, use_fips_endpoint 및 endpoint_url 설정입니다.
8. 각 URL 의 기본 엔드포인트 AWS 서비스 가 마지막으로 사용됩니다. 각 리전에서 사용할 수 있는 표준 서비스 엔드포인트 목록은 Amazon Web Services 일반 참조의 AWS 리전 및 엔드포인트를 참조하세요.

--no-paginate

출력 페이지 매김을 생성하는 모든 명령 결과를 수신하기 위해 이 자동으로 AWS CLI 수행하는 여러 호출을 비활성화하는 부울 스위치입니다. 즉, 출력의 첫 번째 페이지만 표시됩니다.

--no-sign-request

AWS 서비스 엔드포인트에 대한 HTTP 요청에 서명을 비활성화하는 부울 스위치입니다. 이렇게 하면 보안 인증이 로드되는 것을 방지할 수 있습니다.

--no-verify-ssl

기본적으로 는 AWS 서비스와 통신SSL할 때 를 AWS CLI 사용합니다. 각 SSL 연결 및 호출에 대해는 SSL 인증서를 AWS CLI 확인합니다. 이 옵션을 사용하면 SSL 인증서 확인의 기본 동작이 재정의됩니다.

> **Warning**
>
> 이 옵션은 모범 사례가 아닙니다. 를 사용하면 클라이언트와 AWS 서비스 간의 `--no-verify-ssl`트래픽이 더 이상 보호되지 않습니다. 즉, 트래픽은 보안 위험이며 man-in-the-middle악용에 취약합니다. 인증서에 문제가 있는 경우 대신 해당 문제를 해결하는 것이 좋습니다. 인증서 문제 해결 단계는 the section called “SSL 인증서 오류” 섹션을 참조하세요.

--출력 ***<string>***

이 명령에 사용할 출력 형식을 지정합니다. 다음 값 중 하나를 지정할 수 있습니다.

- **`json`** - 출력은 JSON 문자열 형식입니다.
- **`text`** - 출력은 여러 줄의 탭으로 구분된 문자열 값으로 형식이 지정됩니다. 출력을 `grep`, `sed` 또는 `awk`와 같은 텍스트 프로세서로 전달하는 데 사용할 수 있습니다.
- **`table`** - 출력은 셀 테두리를 형성하기 위해 +|- 문자를 사용하여 표로 형식이 지정됩니다. 일반적으로 읽기는 쉽지만 프로그래밍 방식으로는 유용하지 않은 '인간 친화적' 형식으로 정보를 표시합니다.

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

이 명령에 사용할 명명된 프로필을 지정합니다. 명명된 프로필을 추가로 설정하려면 `aws configure` 명령을 `--profile` 옵션과 함께 사용하면 됩니다.

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

--쿼리 ***<string>***

응답 데이터를 필터링하는 데 사용할 JMESPath 쿼리를 지정합니다. 자세한 내용은 에서 출력 필터링 AWS CLI 단원을 참조하십시오.

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

이 명령의 AWS 요청을 보낼 AWS 리전을 지정합니다. 지정할 수 있는 모든 리전 목록은 Amazon Web Services 일반 참조의 AWS 리전 및 엔드포인트를 참조하세요.

--version

실행 중인 AWS CLI 프로그램의 현재 버전을 표시하는 부울 스위치입니다.

## 명령줄 옵션의 일반적인 용도

명령줄 옵션의 일반적인 용도는 AWS 리전에서 리소스를 확인하고 읽기 쉽게 또는 스크립팅할 때 사용하기 쉽게 출력 형식을 변경하는 것입니다. 다음 예제에서는 인스턴스가 있는 리전을 찾을 때까지 각 리전에 대해 describe-instances 명령을 실행합니다.

```
$ 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                                   |||
...
```

# 에서 명령 완료 구성 AWS CLI

AWS Command Line Interface (AWS CLI)에는 탭 키를 사용하여 부분적으로 입력된 명령을 완료할 수 있는 bash 호환 명령 완료 기능이 포함되어 있습니다. 이 기능은 대부분의 시스템에서 수동으로 구성해야 합니다.

주제



## 작동 방식

명령, 파라미터 또는 옵션을 부분적으로 입력하면 명령 완성 기능이 자동으로 명령을 완성하거나 제안된 명령 목록을 표시합니다. 명령 완료를 프롬프트하려면 명령을 부분적으로 입력하고 일반적으로 *Tab* 대부분의 쉘에 있습니다.

다음 예제에서는 명령 완성을 사용할 수 있는 여러 가지 방법을 보여줍니다.

- 명령을 부분적으로 입력하고 를 누릅니다.*Tab* 제안된 명령 목록을 표시합니다.

```
$ 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
```

- 파라미터를 부분적으로 입력하고 를 누릅니다.*Tab* 제안된 파라미터 목록을 표시합니다.

```
$ 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
```

- 파라미터를 입력하고 를 누릅니다.*Tab* 제안된 리소스 값 목록을 표시합니다. 이 기능은 AWS CLI 버전 2에서만 사용할 수 있습니다.

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

## Linux 또는 macOS에서 명령 완성 구성

Linux 또는 macOS에서 명령 완성을 구성하려면 사용 중인 셸의 이름과 `aws_completer` 스크립트의 위치를 알아야 합니다.

> **Note**
>
> 명령 완료는 Amazon Linux를 실행하는 Amazon EC2 인스턴스에서 기본적으로 자동으로 구성되고 활성화됩니다.

주제



### 경로에 completer 폴더가 있는지 확인

AWS 완료자가 성공적으로 작동하려면 가 쉘의 경로에 있어야 `aws_completer` 합니다. `which` 명령을 사용하여 경로에 completer가 있는지 확인할 수 있습니다.

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

이 명령으로 completer를 찾을 수 없는 경우 다음 단계를 사용하여 경로에 completer의 폴더를 추가합니다.

1단계: AWS 완료자 찾기

AWS 완료자의 위치는 사용되는 설치 방법에 따라 달라질 수 있습니다.

- 패키지 관리자 - pip, brew, 및 yum와 같은 프로그램은 apt-get 일반적으로 AWS 완료자(또는 그에 대한 대칭 링크)를 표준 경로 위치에 설치합니다.
  - pip을 --user 파라미터 없이 사용한 경우 기본 경로는 /usr/local/bin/aws_completer입니다.
  - pip을 --user 파리미터와 함께 사용한 경우 기본 경로는 /home/*username*/.local/bin/aws_completer입니다.
- 번들에 포함된 설치 관리자 - 번들에 포함된 설치 관리자를 사용한 경우 기본 경로는 /usr/local/bin/aws_completer입니다.

다른 모든 항목이 실패하면 find 명령을 사용하여 파일 시스템에서 AWS 완료자를 검색할 수 있습니다.

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

2단계: 셸 식별

사용 중인 셸을 식별하려면 다음 명령 중 하나를 사용하면 됩니다.

- 에코 $SHELL - 셸의 프로그램 파일 이름을 표시합니다. 이 항목은 로그인 후 다른 셸을 시작하지 않은 한 일반적으로 사용 중인 셸의 이름과 일치합니다.

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

- ps - 현재 사용자에 대해 실행 중인 프로세스를 표시합니다. 그 중 하나가 셸입니다.

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

3단계: 경로에 completer 추가

1. 사용자 폴더에서 셸의 프로파일 스크립트를 찾습니다.

```
$ ls -a ~/
```

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

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

2. 다음 예제와 유사한 프로필 스크립트 끝에 내보내기 명령을 추가합니다. */usr/local/bin/*을 이전 섹션에서 검색한 폴더의 이름으로 바꿉니다.

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

3. 현재 세션에 프로필을 다시 로드하여 해당 변경 사항을 적용합니다. *.bash_profile*을 첫 단원에서 검색한 shell 스크립트의 이름으로 바꿉니다.

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

## 명령 완성 활성화

completer가 경로에 있는지 확인한 후에는 사용 중인 셸에 적합한 명령을 실행하여 명령 완성을 활성화합니다. 셸의 프로파일에 명령을 추가하여 새 셸을 열 때마다 실행되도록 할 수 있습니다. 각 명령에서 */usr/local/bin/* 에서 시스템에 있는 경로입니다경로에 completer 폴더가 있는지 확인.

- **bash** – 기본 제공 명령인 `complete`를 사용합니다.

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

  `~/.bashrc`에 이전 명령을 추가하여 새 셸을 열 때마다 실행되도록 합니다. `~/.bash_profile`은 `~/.bashrc`를 소스로 하여 로그인 셸에서도 이 명령이 실행되도록 합니다.

- **zsh** - 명령 완성을 실행하려면 `bashcompinit` 프로파일 스크립트 끝에 다음 자동 로드 행을 추가하여 `~/.zshrc`를 실행해야 합니다.

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

  명령 완성을 사용하려면 기본 제공 명령 `complete`를 사용합니다.

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

~/.zshrc에 이전 명령을 추가하여 새 셸을 열 때마다 실행되도록 합니다.

- **tcsh** - tcsh의 경우 단어 유형 및 패턴을 가져와서 완성 동작을 정의하는 방식으로 완성을 수행합니다.

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

~/.tschrc에 이전 명령을 추가하여 새 셸을 열 때마다 실행되도록 합니다.

명령 완성을 활성화한 후 명령 완성이 작동하는지 확인(명령 완성 확인)합니다.

### 명령 완성 확인

명령 완성을 활성화한 후 셸을 다시 로드하고 부분 명령을 입력한 다음 Tab 키를 눌러 사용 가능한 명령을 봅니다.

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

## Windows에서 명령 완료 구성

> **Note**
>
> 가 다양한 완료 키를 포함하여 완료를 PowerShell 처리하는 방법에 대한 자세한 내용은 Microsoft PowerShell Docs 의 about_Tab_Expansion을 참조하세요.

Windows PowerShell 에서 에 대한 명령 완료를 활성화하려면 에서 다음 단계를 완료합니다 PowerShell.

1. 다음 명령을 사용하여 $PROFILE을 엽니다.

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

$PROFILE이 없는 경우 다음 명령을 사용하여 사용자 프로필을 생성합니다.

```
PS C:\> if (!(Test-Path -Path $PROFILE ))
```

```
{ New-Item -Type File -Path $PROFILE -Force }
```

PowerShell 프로파일에 대한 자세한 내용은 Microsoft Docs 웹 사이트의 Windows에서 프로파일을 사용하는 방법을 PowerShell ISE 참조하세요.

2. 명령 완성을 활성화하려면 다음 코드 블록을 프로필에 추가하고 저장한 다음 파일을 닫습니다.

```
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. 명령 완성을 활성화한 후 셸을 다시 로드하고 부분 명령을 입력한 다음 Tab 키를 눌러 사용 가능한 명령을 순환합니다.

```
$ aws sTab
```

```
$ aws s3
```

완성을 위해 사용 가능한 명령을 모두 보려면 부분 명령을 입력하고 Ctrl+Space를 누릅니다.

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

# AWS CLI 에서 재시도 AWS CLI

이 주제에서는 가 예기치 않은 문제로 인해 AWS 서비스 호출이 실패하는 것을 볼 AWS CLI 수 있는 방법을 설명합니다. 이러한 문제는 서버 측에서 발생하거나 호출하려는 AWS 서비스의 속도 제한으로

인해 실패할 수 있습니다. 이러한 종류의 실패는 일반적으로 특별한 처리가 필요하지 않으며 주로 짧은 대기 기간 후에 자동으로 다시 호출됩니다. 는 AWS CLI 이러한 종류의 오류 또는 예외가 발생할 때 AWS 서비스에 대한 클라이언트 호출을 재시도하는 데 도움이 되는 다양한 기능을 제공합니다.

주제



## 사용 가능한 재시도 모드

AWS CLI 에는 버전에 따라 선택할 수 있는 여러 모드가 있습니다.



### 레거시 재시도 모드

레거시 모드는 AWS CLI 버전 1에서 사용하는 기본 모드입니다.  레거시 모드는 다음을 포함하는 제한된 기능을 가진 이전 재시도 핸들러를 사용합니다.

- 최대 재시도 횟수에 대한 기본값은 4이며, 총 5회의 호출을 시도합니다. 이 값은 `max_attempts` 구성 파라미터를 통해 덮어쓸 수 있습니다.
- DynamoDB는 최대 재시도 횟수의 기본값이 9이며, 총 10회의 호출을 시도합니다. 이 값은 `max_attempts` 구성 파라미터를 통해 덮어쓸 수 있습니다.
- 다음과 같은 제한된 수의 오류/예외에 대한 재시도 횟수:
  - 일반 소켓/연결 오류:
    - `ConnectionError`
    - `ConnectionClosedError`
    - `ReadTimeoutError`
    - `EndpointConnectionError`
  - 서비스 측 조절/제한 오류 및 예외:
    - `Throttling`

    - ThrottlingException
    - ThrottledException
    - RequestThrottledException
    - ProvisionedThroughputExceededException
- 429, 500, 502, 503, 504 및 509를 포함한 여러 HTTP 상태 코드에 대해 재시도합니다.
- 모든 재시도 횟수에는 기본 계수 2의 지수 백오프가 포함됩니다.

### 표준 재시도 모드

표준 모드는 레거시보다 더 많은 기능을 가진 의 AWS SDKs 표준 재시도 규칙 집합입니다. 표준 모드는 AWS CLI 버전 2에 대해 생성되었으며 AWS CLI 버전 1로 백포팅됩니다. 표준 모드의 기능은 다음과 같습니다.

- 최대 재시도 횟수에 대한 기본값은 2이며, 총 3회의 호출을 시도합니다. 이 값은 `max_attempts` 구성 파라미터를 통해 덮어쓸 수 있습니다.
- 다음과 같은 확장된 오류/예외 목록에 대한 재시도 횟수:
    - 일시적 오류/예외
        - RequestTimeout
        - RequestTimeoutException
        - PriorRequestNotComplete
        - ConnectionError
        - HTTPClientError
    - 서비스 측 조절/제한 오류 및 예외:
        - Throttling
        - ThrottlingException
        - ThrottledException
        - RequestThrottledException
        - TooManyRequestsException
        - ProvisionedThroughputExceededException
        - TransactionInProgressException
        - RequestLimitExceeded
        - BandwidthLimitExceeded

  - `LimitExceededException`
  - `RequestThrottled`
  - `SlowDown`
  - `EC2ThrottledException`

- 설명적이지 않은 일시적인 오류 코드에 대한 재시도 횟수. 특히 500, 502, 503, 504 HTTP 상태 코드입니다.
- 모든 재시도 횟수에는 최대 백오프 시간 20초 동안 기본 계수 2의 지수 백오프가 포함됩니다.

### 적응형 재시도 모드

> **Warning**
>
> 적응형 모드는 실험적 모드이며 기능 및 동작 모두 변경될 수 있습니다.

적응형 재시도 모드는 표준 모드의 모든 기능을 포함하는 실험적 재시도 모드입니다. 표준 모드 기능 외에도 적응형 모드는 각 재시도 시 동적으로 업데이트되는 토큰 버킷 및 속도 제한 변수를 사용하여 클라이언트 측 속도 제한도 도입합니다. 이 모드는 AWS 서비스의 오류/예외 상태 응답에 적응하는 클라이언트 측 재시도에 유연성을 제공합니다.

새로 재시도할 때마다 적응 모드는 AWS 서비스 응답에 표시된 오류, 예외 또는 HTTP 상태 코드를 기반으로 속도 제한 변수를 수정합니다. 이러한 속도 제한 변수는 클라이언트의 새 호출 속도를 계산하는 데 사용됩니다. 서비스의 각 예외/오류 또는 비성공 HTTP 응답(위 목록에 제공됨) AWS 은 성공하거나 토큰 버킷이 소진되거나 구성된 최대 시도 횟수 값에 도달할 때까지 재시도가 발생할 때 속도 제한 변수를 업데이트합니다.

## 재시도 모드 구성

에는 클라이언트 객체를 생성할 때 고려해야 할 다양한 재시도 구성과 구성 방법이 모두 AWS CLI 포함되어 있습니다.

### 사용 가능한 구성 방법

에서 AWS CLI사용자는 다음과 같은 방법으로 재시도를 구성할 수 있습니다.

- 환경 변수

- AWS CLI 구성 파일

사용자는 다음 재시도 옵션을 사용자 지정할 수 있습니다.

- 재시도 모드 - 에서 AWS CLI 사용하는 재시도 모드를 지정합니다. 앞에서 설명한 대로 레거시, 표준 및 적응형, 이렇게 3가지 재시도 모드를 사용할 수 있습니다. AWS CLI 버전 1의 기본값은 레거시 .
- 최대 시도 횟수 - AWS CLI 재시도 핸들러가 사용하는 최대 재시도 횟수의 값을 지정합니다. 여기서 초기 호출은 사용자가 제공하는 값으로 계산됩니다. 기본값은 5입니다.

### 환경 변수에서 재시도 구성 정의

에 대한 재시도 구성을 정의하려면 운영 체제의 환경 변수를 AWS CLI업데이트합니다.

재시도 환경 변수는 다음과 같습니다.

- `AWS_RETRY_MODE`
- `AWS_MAX_ATTEMPTS`

환경 변수에 대한 자세한 내용은 에 대한 환경 변수 구성 AWS CLI 섹션을 참조하세요.

## 재시도 로그 보기

는 Boto3의 재시도 방법론 및 로깅을 AWS CLI 사용합니다. 모든 명령에서 `--debug` 옵션을 사용하여 디버그 로그를 받을 수 있습니다. `--debug` 옵션을 사용하는 방법에 대한 자세한 내용은 의 명령줄 옵션 AWS CLI 섹션을 참조하세요.

디버그 로그에서 “재시도”를 검색하면 필요한 재시도 정보를 찾을 수 있습니다. 재시도를 위한 클라이언트 로그 항목은 활성화한 재시도 모드에 따라 다릅니다.

레거시 모드:

재시도 메시지는 botocore.retryhandler에 의해 생성됩니다. 다음 3가지 메시지 중 하나가 표시됩니다.

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

표준 또는 적응형 모드:

재시도 메시지는 botocore.retries.standard에 의해 생성됩니다. 다음 3가지 메시지 중 하나가 표시됩니다.

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

botocore 재시도의 전체 정의 파일은 botocore GitHub 리포지토리의 _retry.json을 참조하세요.

# 에 대한 HTTP 프록시 사용 AWS CLI

프록시 서버를 AWS 통해 에 액세스하려면 프록시 서버에서 사용하는 DNS 도메인 이름 또는 IP 주소 및 포트 번호를 사용하여 HTTP_PROXY 및 HTTPS_PROXY 환경 변수를 구성할 수 있습니다.

주제



## 예제 사용

> **Note**
>
> 다음 예제에서는 환경 변수 이름을 모두 대문자로 표시합니다. 그러나 다른 대소문자를 사용하여 변수를 두 번 지정하는 경우 소문자가 우선합니다. 시스템 혼란과 예상하지 못한 동작을 피하기 위해 각 변수를 한 번만 정의하는 것이 좋습니다.

다음 예제에서는 프록시의 명시적 IP 주소 또는 프록시의 IP 주소로 확인되는 DNS 이름을 사용하는 방법을 보여줍니다. 어떤 경우든 콜론과 쿼리가 전송되는 포트 이름이 뒤에 나올 수 있습니다.

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

모든 세션에 대해 설정하려면

```
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
```

환경 변수를 설정하는 데 setx를 사용하면 현재 명령 프롬프트 세션과 명령 실행 후 생성한 모든 명령 프롬프트 세션에서 사용되는 값이 변경됩니다. 명령을 실행하는 시점에 이미 실행 중인 다른 명령 쉘에는 영향을 주지 않습니다.

현재 세션에만 설정하려면

환경 변수를 설정하는 데 set을 사용하면 사용되는 값이 변경되어 현재 명령 프롬프트 세션이 종료될 때까지 또는 변수를 다른 값으로 설정할 때까지 유지됩니다.

```
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
```

## 프록시에 인증

> **Note**
>
> AWS CLI 는 NTLM 프록시를 지원하지 않습니다. NTLM 또는 Kerberos 프로토콜 프록시를 사용하는 경우 Cntlm과 같은 인증 프록시를 통해 연결할 수 있습니다.

는 HTTP 기본 인증을 AWS CLI 지원합니다. 다음과 URL같이 프록시 에서 사용자 이름과 암호를 지정합니다.

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

모든 세션에 대해 설정하려면

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

현재 세션에만 설정하려면

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

## Amazon EC2 인스턴스에서 프록시 사용

연결된 IAM 역할로 시작된 Amazon EC2 인스턴스에서 프록시를 구성하는 경우 인스턴스 메타데이터에 액세스하는 데 사용되는 주소를 면제해야 합니다. 이렇게 하려면 NO_PROXY 환경 변수를 인스턴스 메타데이터 서비스의 IP 주소 169.254.169.254로 설정합니다. 이 주소는 달라지지 않습니다.

Linux or macOS

```
$ export NO_PROXY=169.254.169.254
```

Windows Command Prompt

모든 세션에 대해 설정하려면

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

현재 세션에만 설정하려면

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

## 문제 해결

에 문제가 발생하면 문제 해결 단계는 섹션을 AWS CLI참조오류 해결하세요. 가장 관련성이 높은 문제 해결 단계는 the section called “SSL 인증서 오류” 섹션을 참조하세요.

# 에서 엔드포인트 사용 AWS CLI

에 프로그래밍 방식으로 연결하려면 엔드포인트를 AWS 서비스사용합니다. 엔드포인트는 AWS 웹 서비스에 대한 진입점URL의 입니다. AWS Command Line Interface (AWS CLI)는 의 각 서비스에 대해 기본 엔드포인트를 자동으로 사용하지 AWS 리전만 API 요청에 대해 대체 엔드포인트를 지정할 수 있습니다.

엔드포인트 주제



## 단일 명령에 대한 엔드포인트 설정

단일 명령에 대한 엔드포인트 설정이나 환경 변수를 재정의하려면 `--endpoint-url` 명령줄 옵션을 사용하세요. 다음 명령 예제에서는 사용자 지정 Amazon S3 엔드포인트 를 사용합니다URL.

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

## 모든 에 대한 전역 엔드포인트 설정 AWS 서비스

모든 서비스에 대한 요청을 사용자 지정 엔드포인트 로 라우팅하려면 다음 설정 중 하나를 URL사용합니다.

- 환경 변수:
  - `AWS_IGNORE_CONFIGURED_ENDPOINT_URLS` - 구성된 엔드포인트를 무시합니다URLs.

    Linux or macOS

    ```
    $ export AWS_IGNORE_CONFIGURED_ENDPOINT_URLS=true
    ```

Windows Command Prompt

모든 세션에 대해 설정하려면

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

현재 세션에만 설정하려면

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

PowerShell

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

- AWS_ENDPOINT_URL - 전역 엔드포인트를 설정합니다URL.

  Linux or macOS

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

  Windows Command Prompt

  모든 세션에 대해 설정하려면

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

  현재 세션에만 설정하려면

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

  PowerShell

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

- config 파일:

  - ignore_configure_endpoint_urls - 구성된 엔드포인트를 무시합니다URLs.

    ```
    ignore_configure_endpoint_urls = true
    ```

- endpoint_url - 전역 엔드포인트를 설정합니다URL.

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

서비스별 엔드포인트와 --endpoint-url 명령줄 옵션은 모든 전역 엔드포인트를 재정의합니다.

## 모든 에 대해 FIPs 엔드포인트를 사용하도록 설정 AWS 서비스

FIPs 엔드포인트를 사용할 모든 서비스에 대한 요청을 라우팅하려면 다음 중 하나를 사용합니다.

- AWS_USE_FIPS_ENDPOINT 환경 변수

  Linux or macOS

  ```
  $ export AWS_USE_FIPS_ENDPOINT=true
  ```

  Windows Command Prompt

  모든 세션에 대해 설정하려면

  ```
  C:\> setx AWS_USE_FIPS_ENDPOINT true
  ```

  현재 세션에만 설정하려면

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

  PowerShell

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

- use_fips_endpoint 파일 설정

```
use_fips_endpoint = true
```

일부 AWS 서비스는 일부 에서 연방 정보 처리 표준(FIPS) 140-2를 지원하는 엔드포인트를 제공합니다 AWS 리전. AWS 서비스가 를 지원하는 경우 FIPS이 설정은 에서 를 사용해야 AWS CLI 하는 FIPS 엔드포인트를 지정합니다. 표준 AWS 엔드포인트와 달리 FIPS 엔드포인트는 FIPS 140-2를 준수하는

TLS 소프트웨어 라이브러리를 사용합니다. 이러한 엔드포인트는 미국 정부와 상호 작용하는 기업에 필요할 수 있습니다.

이 설정이 활성화되었지만 의 서비스에 대한 FIPS 엔드포인트가 없는 경우 AWS 명령 AWS 리전이 실패할 수 있습니다. 이 경우 `--endpoint-url` 옵션을 사용하여 명령에 사용할 엔드포인트를 수동으로 지정하거나 서비스별 엔드포인트를 사용합니다.

로 FIPS 엔드포인트를 지정하는 방법에 대한 자세한 내용은 서비스별 엔드포인트 를 AWS 리전참조하세요. FIPS

## 모든 AWS 서비스에 이중 스택 엔드포인트를 사용하도록 설정

모든 서비스에 대한 요청을 사용 가능한 이중 스택 엔드포인트로 라우팅하려면 다음 설정 중 하나를 사용하세요.

- `AWS_USE_DUALSTACK_ENDPOINT` 환경 변수

  Linux or macOS

  ```
  $ export AWS_USE_DUALSTACK_ENDPOINT=true
  ```

  Windows Command Prompt

  모든 세션에 대해 설정하려면

  ```
  C:\> setx AWS_USE_DUALSTACK_ENDPOINT true
  ```

  현재 세션에만 설정하려면

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

  PowerShell

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

- `use_dualstack_endpoint` 파일 설정

  ```
  use_dualstack_endpoint = true
  ```

듀얼 스택 엔드포인트를 사용하여 AWS 요청을 보낼 수 있습니다. IPv4 및 IPv6 트래픽을 모두 지원하는 듀얼 스택 엔드포인트에 대한 자세한 내용은 Amazon Simple Storage Service 사용 설명서의 Amazon S3 듀얼 스택 엔드포인트 사용을 참조하세요. 이중 스택 엔드포인트는 일부 리전에 사용할 수 있는 서비스입니다. 서비스 또는 에 대한 듀얼 스택 엔드포인트가 없는 경우 요청이 실패 AWS 리전 합니다. 이 옵션은 기본적으로 비활성화되어 있습니다.

## 서비스별 엔드포인트 설정

서비스별 엔드포인트 구성은 AWS CLI 요청에 대해 선택한 영구 엔드포인트를 사용하는 옵션을 제공합니다. 이러한 설정은 로컬 엔드포인트, VPC 엔드포인트 및 타사 로컬 AWS 개발 환경을 지원할 수 있는 유연성을 제공합니다. 테스트 환경과 프로덕션 환경에 서로 다른 엔드포인트를 사용할 수 있습니다. 개별 에 URL 대한 엔드포인트를 지정할 수 있습니다 AWS 서비스.

서비스별 엔드포인트는 다음과 같은 방법으로 지정할 수 있습니다.

- 단일 명령에 대한 명령줄 옵션 `--endpoint-url`.
- 환경 변수:
  - `AWS_IGNORE_CONFIGURED_ENDPOINT_URLS` - 명령줄에 지정URLs되지 않은 한 구성된 모든 엔드포인트를 무시합니다.
  - `AWS_ENDPOINT_URL_<SERVICE>` - 특정 서비스에 사용되는 사용자 지정 엔드포인트를 지정하며, 여기서 <SERVICE>는 AWS 서비스 서비스 식별자로 대체됩니다. 모든 서비스별 변수에 대해서는 the section called “서비스별 식별자 목록”를 참조하세요
- `config` 파일:
  - `ignore_configure_endpoint_urls` - 환경 변수를 사용하거나 명령줄에 지정URLs하지 않는 한 구성된 모든 엔드포인트를 무시합니다.
  - `config` 파일의 `services` 섹션과 `endpoint_url` 파일 설정이 결합됩니다.

서비스별 엔드포인트 주제:

- 서비스별 엔드포인트: 환경 변수
- 서비스별 엔드포인트: 공유 config 파일
- 서비스별 엔드포인트: 서비스별 식별자 목록

### 서비스별 엔드포인트: 환경 변수

환경 변수는 구성 파일의 설정을 재정의하지만 명령줄에 지정된 옵션을 재정의하지는 않습니다. 모든 프로파일이 디바이스에서 동일한 엔드포인트를 사용하도록 하려면 환경 변수를 사용하세요.

다음은 서비스별 환경 변수입니다.

- AWS_IGNORE_CONFIGURED_ENDPOINT_URLS - 명령줄에 지정URLs되지 않은 한 구성된 모든 엔드포인트를 무시합니다.

  Linux or macOS

  ```
  $ export AWS_IGNORE_CONFIGURED_ENDPOINT_URLS=true
  ```

  Windows Command Prompt

  모든 세션에 대해 설정하려면

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

  현재 세션에만 설정하려면

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

  PowerShell

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

- AWS_ENDPOINT_URL_<SERVICE> - 특정 서비스에 사용되는 사용자 지정 엔드포인트를 지정합니다. 여기서 <SERVICE> 는 AWS 서비스 식별자로 대체됩니다. 모든 서비스별 변수에 대해서는 the section called “서비스별 식별자 목록”를 참조하세요

  다음 환경 변수 예제는 AWS Elastic Beanstalk의 엔드포인트를 설정합니다.

  Linux or macOS

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

  Windows Command Prompt

  모든 세션에 대해 설정하려면

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

  현재 세션에만 설정하려면

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

PowerShell

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

환경 변수 설정에 대한 자세한 내용은 환경 변수를 사용하여 the section called “환경 변수”을 참조하세요.

### 서비스별 엔드포인트: 공유 **config** 파일

공유 `config` 파일에서 `endpoint_url`은 여러 섹션에서 사용됩니다. 서비스별 엔드포인트를 설정하려면 `services` 섹션 내의 서비스 식별자 키 아래에 중첩된 `endpoint_url` 설정을 사용하세요. 공유 `config` 파일에서 `services` 섹션을 정의하는 방법에 대한 자세한 내용은 the section called “`services`”를 참조하세요.

다음 예제에서는 `services` 섹션을 사용하여 Amazon S3에 URL 대한 서비스별 엔드포인트와 다른 모든 서비스에 사용되는 사용자 지정 글로벌 엔드포인트를 구성합니다.

```
[profile dev1]
endpoint_url = http://localhost:1234
services = s3-specific

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

단일 프로파일로 여러 서비스에 대한 엔드포인트를 구성할 수 있습니다. 다음 예제에서는 Amazon S3 및 에 URLs 대한 서비스별 엔드포인트를 동일한 프로파일 AWS Elastic Beanstalk 에 설정합니다.

`services` 섹션에서 사용할 모든 서비스 식별자 키 목록은 서비스별 식별자 목록을 참조하세요.

```
[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
```

서비스 구성 섹션은 여러 프로파일에서 사용할 수 있습니다. 다음 예제에서는 두 개의 프로파일이 동일한 services 정의를 사용합니다.

```
[profile dev1]
output = json
services = testing-s3

[profile dev2]
output = text
services = testing-s3

[services testing-s3]
s3 =
  endpoint_url = https://localhost:4567
```

### 서비스별 엔드포인트: 서비스별 식별자 목록

AWS 서비스 식별자는 모든 공백을 밑줄로 바꾸고 모든 문자를 소문자로 표시serviceId하여 API 모델의 를 기반으로 합니다.

다음 서비스 식별자 예제에서는 를 사용합니다 AWS Elastic Beanstalk. AWS Elastic Beanstalk 의 는 serviceId Elastic Beanstalk이므로 서비스 식별자 키는 입니다elastic_beanstalk.

다음 표에는 모든 서비스별 식별자, config 파일 키 및 환경 변수가 나열되어 있습니다.

## 엔드포인트 구성 및 설정 우선 순위

엔드포인트 구성 설정은 시스템 또는 사용자 환경 변수, 로컬 AWS 구성 파일 또는 명령줄에 파라미터로 명시적으로 선언되는 등 여러 위치에 있습니다. AWS CLI 엔드포인트 구성 설정은 다음 순서에 따라 우선적으로 적용됩니다.

1. --endpoint-url 명령줄 옵션
2. 사용 설정된 경우, 사용자 지정 엔드포인트를 무시하도록 AWS_IGNORE_CONFIGURED_ENDPOINT_URLS 글로벌 엔드포인트 환경 변수 또는 프로파일 설정 ignore_configure_endpoint_urls를 사용합니다.
3. 서비스별 환경 변수 AWS_ENDPOINT_URL_<SERVICE>에서 제공하는 값(예: AWS_ENDPOINT_URL_DYNAMODB).

4. AWS_USE_DUALSTACK_ENDPOINT, AWS_USE_FIPS_ENDPOINT 및 AWS_ENDPOINT_URL 환경 변수에서 제공하는 값입니다.
5. 공유 config 파일의 services 섹션 내의 endpoint_url 설정에서 제공하는 서비스별 엔드포인트 값.
6. 공유 config 파일의 profile 내에서 endpoint_url 설정에 의해 제공되는 값.
7. use_dualstack_endpoint, use_fips_endpoint 및 endpoint_url 설정입니다.
8. 각 URL 의 기본 엔드포인트 AWS 서비스 가 마지막으로 사용됩니다. 각 리전에서 사용할 수 있는 표준 서비스 엔드포인트 목록은 Amazon Web Services 일반 참조의 AWS 리전 및 엔드포인트를 참조하세요.

# 에 대한 인증 및 액세스 자격 증명 AWS CLI

서비스로 개발할 AWS 때 가 를 AWS CLI 인증하는 AWS 방법을 설정해야 합니다. 에 대한 프로그래밍 방식 액세스에 대한 자격 증명을 구성하려면 다음 옵션 중 하나를 AWS CLI선택합니다. 옵션은 권장 순서입니다.

| 프로그래밍 방식 액세스가 필요한 사용자는 누구인가요? | 용도 | 지침 |
| --- | --- | --- |
| IAM | 단기 보안 인증 정보를 사용합니다. | the section called “단기 보안 인증” |
| IAM | 역할을 보안 인증으로 사용합니다. | the section called “IAM 역할” |
| IAM | (권장되지 않음)장기 보안 인증 정보를 사용합니다. | the section called “IAM 사용자” |

## 구성 및 보안 인증 우선 순위

자격 증명 및 구성 설정은 시스템 또는 사용자 환경 변수, 로컬 AWS 구성 파일 또는 명령줄에 파라미터로 명시적으로 선언되는 등 여러 위치에 있습니다. 특정 위치가 다른 위치보다 우선합니다. AWS CLI 보안 인증 및 구성 설정은 다음 순서에 따라 우선적으로 적용됩니다.

1. 명령줄 옵션 - `--region`, `--output`, `--profile`와 같은 다른 위치의 설정을 재정의합니다.
2. 환경 변수 - 시스템의 환경 변수에 값을 저장할 수 있습니다.
3. 역할 수임 - 구성 또는 `aws sts assume-role` 명령을 통해 IAM 역할의 권한을 수임합니다.
4. 웹 ID로 역할 수임 - 구성 또는 `aws sts assume-role` 명령을 통해 웹 ID를 사용하여 IAM 역할의 권한을 수임합니다.
5. 보안 인증 파일 - `aws configure` 명령을 실행하면 `credentials` 및 `config` 파일이 업데이트됩니다. `credentials` 파일은 `~/.aws/credentials`(Linux 또는 macOS) 또는 `C:\Users\`*`USERNAME`*`\.aws\credentials`(Windows)에 저장됩니다.
6. 사용자 지정 프로세스 - 외부 소스에서 보안 인증을 가져옵니다.

7. 구성 파일 - `aws configure` 명령을 실행하면 `credentials` 및 `config` 파일이 업데이트됩니다. `config` 파일은 `~/.aws/config`(Linux 또는 macOS) 또는 `C:\Users\USERNAME\.aws\config`(Windows)에 저장됩니다.
8. 컨테이너 보안 인증 - IAM 역할을 각 Amazon Elastic Container Service(AmazonECS) 태스크 정의와 연결할 수 있습니다. 그러면 작업의 컨테이너에 대해 해당 역할의 임시 보안 인증을 사용할 수 있습니다. 자세한 내용은 Amazon Elastic Container Service 개발자 안내서의 IAM 작업에 대한 역할을 참조하세요.
9. Amazon EC2 인스턴스 프로필 자격 증명 - IAM 역할을 각 Amazon Elastic Compute Cloud(AmazonEC2) 인스턴스와 연결할 수 있습니다. 그러면 인스턴스에서 실행되는 코드에 대해 해당 역할의 임시 보안 인증을 사용할 수 있습니다. 자격 증명은 Amazon EC2 메타데이터 서비스를 통해 전달됩니다. 자세한 내용은 IAM Amazon 사용 설명서EC2의 Amazon EC2 역할 및 IAM 사용 설명서의 인스턴스 프로파일 사용을 참조하세요.

## 이 섹션의 추가 주제

- the section called “단기 보안 인증”
- the section called “IAM 역할”
- the section called “IAM 사용자”
- the section called “에서 Amazon EC2 인스턴스 메타데이터를 보안 인증으로 사용 AWS CLI”
- the section called “외부 자격 증명”

## 에 대한 단기 자격 증명으로 인증 AWS CLI

확장된 세션 기간 옵션과 함께 IAM Identity Center 인증을 사용하도록 SDK 또는 도구를 구성하는 것이 좋습니다. 그러나 AWS 액세스 포털에서 사용할 수 있는 임시 보안 인증 정보를 복사하고 사용할 수 있습니다. 보안 인증이 만료되면 새 보안 인증을 복사해야 합니다. 프로파일에서 임시 보안 인증을 사용하거나 이를 시스템 속성 및 환경 변수의 값으로 사용할 수 있습니다.

1. AWS 액세스 포털 에 로그인합니다.
2. 다음 지침에 따라 AWS 액세스 포털에서 IAM 역할 자격 증명을 복사합니다.

    1. 연결된 지침의 2단계에서 개발 요구 사항에 대한 액세스 권한을 부여하는 AWS 계정 및 IAM 역할 이름을 선택합니다. 이 역할에는 일반적으로 PowerUserAccess 또는 개발자 와 같은 이름이 있습니다.
    2. 4단계에서 AWS 보안 인증 파일에 프로필 추가 옵션을 선택하고 내용을 복사합니다.

3. 공유 `credentials` 파일을 생성하거나 엽니다. 이 파일은 Linux 및 macOS 시스템의 경우 `~/.aws/credentials`이며, Windows의 경우 `%USERPROFILE%\.aws\credentials`입니다. 자세한 내용은 the section called “의 구성 및 보안 인증 파일 설정 AWS CLI” 단원을 참조하십시오.
4. 다음 텍스트를 공유 `credentials` 파일에 추가합니다. 샘플 값을 복사한 보안 인증으로 바꿉니다.

```
[default]
aws_access_key_id = AKIAIOSFODNN7EXAMPLE
aws_secret_access_key = wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
aws_session_token =
 IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZ2luX2IQoJb3JpZVERYLONGSTRINGEXAMPLE
```

5. 원하는 기본 지역 및 형식을 공유 `config` 파일에 추가합니다.

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

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

가 서비스 클라이언트를 SDK 생성하면 이러한 임시 자격 증명에 액세스하여 각 요청에 사용합니다. 2a단계에서 선택한 IAM 역할에 대한 설정은 임시 보안 인증이 유효한 기간을 결정합니다. 최대 유효 기간은 12시간입니다.

보안 인증 정보가 만료될 때마다 이 단계를 반복하세요.

# 에서 IAM 역할 사용 AWS CLI

AWS Identity and Access Management (IAM) 역할은 사용자가 추가(또는 다른) 권한을 얻거나 다른 AWS 계정에서 작업을 수행할 수 있는 권한을 얻을 수 있는 권한 부여 도구입니다.

주제

- 사전 조건
- IAM 역할 사용 개요
- 역할 구성 및 사용

- 멀티 팩터 인증 사용
- 교차 계정 역할 및 외부 ID
- 보다 쉬운 감사를 위한 역할 세션 이름 지정
- 웹 자격 증명을 사용한 역할 수임
- 캐시된 자격 증명 지우기

## 사전 조건

`iam` 명령을 실행하려면 AWS CLI를 설치하고 구성해야 합니다. 자세한 내용은 설치 AWS CLI 단원을 참조하십시오.

## IAM 역할 사용 개요

`~/.aws/config` 파일에서 IAM 역할에 대한 프로파일을 정의하여 역할을 사용하도록 AWS Command Line Interface (AWS CLI)를 구성할 수 있습니다.

다음 예제는 `marketingadmin`라는 이름의 역할 프로파일을 보여줍니다. 를 사용하여 명령을 실행하는 경우`--profile marketingadmin`(또는 AWS_PROFILE 환경 변수 로 명령을 지정하는 경우)는 별도의 프로파일에 정의된 보안 인증을 AWS CLI 사용하여 Amazon 리소스 이름(ARN) 로 역할을 `user1` 수임합니다`arn:aws:iam::123456789012:role/marketingadminrole`. 해당 역할에 할당된 권한에서 허용되는 모든 작업을 실행할 수 있습니다.

```
[profile marketingadmin]
role_arn = arn:aws:iam::123456789012:role/marketingadminrole
source_profile = user1
```

그런 다음 역할을 사용할 권한과 함께 사용자 보안 인증 정보가 포함된 별도의 명명된 프로파일을 가리키는 `source_profile`을 지정하면 됩니다. 앞의 예제에서는 `marketingadmin` 프로파일이 `user1` 프로파일의 보안 인증을 사용하고 있습니다. AWS CLI 명령을 프로파일 를 사용하도록 지정하면 는 연결된 `user1` 프로파일의 보안 인증 정보를 `marketingadmin` AWS CLI 자동으로 검색하고 이를 사용하여 지정된 IAM 역할에 대한 임시 보안 인증을 요청합니다. 는 백그라운드에서 sts:AssumeRole 작업을 CLI 사용하여 이를 수행합니다. 이러한 임시 자격 증명은 요청된 AWS CLI 명령을 실행하는 데 사용됩니다. 지정된 역할에는 요청된 AWS CLI 명령을 실행할 수 있는 IAM 권한 정책이 연결되어 있어야 합니다.

Amazon Elastic Compute Cloud(AmazonEC2) 인스턴스 또는 Amazon Elastic Container Service(AmazonECS) 컨테이너 내에서 AWS CLI 명령을 실행하려면 인스턴스 프로파일 또는 컨테이

너에 연결된 IAM 역할을 사용할 수 있습니다. 프로파일을 지정하지 않거나 환경 변수를 설정하지 않은 경우 해당 역할이 직접 사용됩니다. 이렇게 하면 인스턴스에서 수명이 긴 액세스 키를 저장하는 것을 피할 수 있습니다. 또한 이러한 인스턴스 또는 컨테이너 역할을 다른 역할에 대한 자격 증명을 가져오는 데에만 사용할 수 있습니다. 이를 위해서는 `credential_source`(`source_profile` 대신에)를 사용하여 자격 증명을 찾는 방법을 지정해야 합니다. `credential_source` 속성은 다음과 같은 값들을 지원합니다.

- `Environment` - 환경 변수에서 소스 자격 증명을 검색합니다.
- `Ec2InstanceMetadata` - Amazon EC2 인스턴스 프로파일에 연결된 IAM 역할을 사용합니다.
- `EcsContainer` - Amazon ECS 컨테이너에 연결된 IAM 역할을 사용합니다.

다음 예제에서는 Amazon EC2 인스턴스 프로파일을 참조하는 데 사용된 것과 동일한 `marketingadminrole` 역할을 보여줍니다.

```
[profile marketingadmin]
role_arn = arn:aws:iam::123456789012:role/marketingadminrole
credential_source = Ec2InstanceMetadata
```

역할을 호출할 때 멀티 팩터 인증 및 외부 ID(타사에서 클라이언트 리소스에 액세스하는 데 사용)와 같은 추가 옵션이 필요할 수 있습니다. AWS CloudTrail 로그에서 더 쉽게 감사할 수 있는 고유한 역할 세션 이름을 지정할 수도 있습니다.

## 역할 구성 및 사용

IAM 역할을 지정하는 프로파일을 사용하여 명령을 실행하면 는 소스 프로파일의 자격 증명을 AWS CLI 사용하여 AWS Security Token Service (AWS STS)를 호출하고 지정된 역할에 대한 임시 자격 증명을 요청합니다. 원본 프로파일의 사용자에는 지정된 프로파일의 역할에 대한 `sts:assume-role`을 호출할 권한이 있어야 합니다. 이 역할에는 소스 프로파일의 사용자가 역할을 사용할 수 있도록 허용하는 신뢰 관계가 있어야 합니다. 역할에 대한 임시 자격 증명을 가져온 다음 사용하는 프로세스는 종종 역할 수임이라고 합니다.

AWS Identity and Access Management 사용 설명서의 사용자에게 권한을 위임할 역할 생성의 절차에 따라 사용자가 수임할 권한을 IAM 사용하여 에서 역할을 생성할 수 있습니다. IAM 역할과 원본 프로필의 사용자가 동일한 계정에 있는 경우 역할의 신뢰 관계를 구성할 때 자신의 계정 ID를 입력할 수 있습니다.

역할을 생성한 후 사용자가 해당 역할을 수임할 수 있도록 신뢰 관계를 수정합니다.

아래 예제는 역할에 연결할 수 있는 신뢰 정책을 보여줍니다. 이 정책은 만약 해당 계정의 관리자가 사용자에게 `sts:AssumeRole` 권한을 명시적으로 부여하면 123456789012 계정에서 모든 사용자가 역할을 수임하도록 허용합니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:root"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

신뢰 정책은 실제로 권한을 부여하지 않습니다. 계정의 관리자는 정책을 적절한 권한에 연결하여 역할을 수임할 권한을 개별 사용자에게 위임해야 합니다. 아래 예제는 사용자가 `marketingadminrole` 역할만 수임하도록 허용하기 위해 사용자에게 연결할 수 있는 정책을 보여줍니다. 사용자에게 역할을 수임할 수 있는 액세스 권한을 부여하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 역할을 전환할 수 있는 사용자 권한 부여를 참조하세요.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": "arn:aws:iam::123456789012:role/marketingadminrole"
    }
  ]
}
```

사용자는 역할 프로파일을 사용하여 AWS CLI 명령을 실행하는 데 추가 권한이 필요하지 않습니다. 대신 명령을 실행하는 데 필요한 권한은 역할에 연결된 권한으로부터 나옵니다. 역할에 권한 정책을 연결하여 어떤 AWS 리소스에 대해 수행할 수 있는 작업을 지정합니다. (사용자와 동일하게 작동하는) 역할에 권한을 연결하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 IAM 사용자 권한 변경을 참조하세요.

이제 역할 프로파일, 역할 권한, 역할 신뢰 관계 및 사용자 권한이 올바르게 구성되었으므로 `--profile` 옵션을 호출하여 명령줄에서 역할을 사용할 수 있습니다. 예를 들어, 다음은 이 주제의 시작 부분에 있는 예제에서 정의된 대로 `ls` 역할에 연결된 권한을 사용하여 Amazon S3 `marketingadmin` 명령을 호출합니다.

```
$ aws s3 ls --profile marketingadmin
```

여러 호출에 역할을 사용하려면 명령줄에서 현재 세션에 대한 `AWS_PROFILE` 환경 변수를 설정하면 됩니다. 환경 변수를 정의하는 동안 각 명령에서 `--profile` 옵션을 지정할 필요가 없습니다.

Linux 또는 macOS

```
$ export AWS_PROFILE=marketingadmin
```

Windows

```
C:\> setx AWS_PROFILE marketingadmin
```

사용자 및 역할 구성에 대한 자세한 내용은 IAM 사용 설명서의 IAM 자격 증명(사용자, 사용자 그룹 및 역할) 및 IAM 역할을 참조하세요.

## 멀티 팩터 인증 사용

추가 보안을 위해 사용자가 역할 프로파일을 사용하여 전화를 걸려고 할 때 다중 인증(MFA) 디바이스, U2F 디바이스 또는 모바일 앱에서 생성된 일회용 키를 제공해야 할 수 있습니다.

먼저 IAM 역할의 신뢰 관계를 로 수정하도록 선택할 수 있습니다MFA. 이렇게 하면 를 사용하여 먼저 인증하지 않고도 누구나 역할을 사용할 수 없습니다MFA. 해당하는 예는 다음 예제의 `Condition` 행을 참조하십시오. 이 정책은 라는 사용자가 정책을 연결하는 역할을 수임anika할 수 있도록 허용하지만, 를 사용하여 인증하는 경우에만 허용됩니다MFA.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": { "AWS": "arn:aws:iam::123456789012:user/anika" },
      "Action": "sts:AssumeRole",
```

```
      "Condition": { "Bool": { "aws:multifactorAuthPresent": true } }
    }
  ]
}
```

그런 다음 사용자 MFA 디바이스의 를 지정하는 줄을 역할 프로파일ARN에 추가합니다. 다음 샘플 config 파일 항목은 anika라는 사용자가 cli-role 역할에 대한 임시 보안 인증 정보를 요청하기 위해 액세스 키를 사용하는 두 가지 역할 프로파일을 보여줍니다. 사용자 anika는 역할을 맡을 권한이 있으며, 이러한 권한은 역할의 신뢰 정책에서 부여합니다.

```
[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
```

mfa_serial 설정에 그림과 ARN같이 또는 하드웨어 MFA 토큰의 일련 번호가 필요할 수 있습니다.

첫 번째 프로필인 에는 role-without-mfa가 필요하지 않습니다MFA. 그러나 역할에 연결된 이전 신뢰 정책 예제에는 가 필요하므로 이 프로파일로 명령을 실행하려는 MFA시도는 실패합니다.

```
$ aws iam list-users --profile role-without-mfa

An error occurred (AccessDenied) when calling the AssumeRole operation: Access denied
```

두 번째 프로필 항목인 는 사용할 MFA 디바이스를 role-with-mfa식별합니다. 사용자가 이 프로파일로 AWS CLI 명령을 실행하려고 하면 는 MFA 디바이스에서 제공하는 일회용 암호(OTP)를 입력하라는 AWS CLI 메시지를 표시합니다. MFA 인증에 성공하면 명령이 요청된 작업을 수행합니다. OTP 는 화면에 표시되지 않습니다.

```
$ aws iam list-users --profile role-with-mfa
```

```
Enter MFA code for arn:aws:iam::123456789012:mfa/cli-user:
{
    "Users": [
        {
            ...
```

## 교차 계정 역할 및 외부 ID

역할을 교차 계정 역할로 구성하면 사용자가 다른 계정에 속한 역할을 사용할 수 있습니다. 역할 생성 중에 IAM 사용자에게 권한을 위임할 역할 생성 에 설명된 대로 역할 유형을 다른 AWS 계정으로 설정합니다. 필요에 따라 필수 MFA를 선택합니다. 에 MFA 설명된 대로 가 신뢰 관계에서 적절한 조건을 구성합니다멀티 팩터 인증 사용.

계정 전체의 역할을 사용할 수 있는 사용자에 대한 추가 제어를 제공하기 위해 외부 ID를 사용하는 경우 역할 프로파일에 `external_id` 파라미터도 추가해야 합니다. 일반적으로 회사 또는 조직 외부에 있는 사람이 다른 계정을 제어하는 경우에만 이를 사용합니다.

```
[profile crossaccountrole]
role_arn = arn:aws:iam::234567890123:role/SomeRole
source_profile = default
mfa_serial = arn:aws:iam::123456789012:mfa/saanvi
external_id = 123456
```

## 보다 쉬운 감사를 위한 역할 세션 이름 지정

한 역할을 여러 개인이 공유하는 경우 감사가 더 어려워집니다. 호출된 각 작업을 호출한 개인에 연결하려고 합니다. 그러나 개인이 역할을 사용할 때 개인에 의한 역할 수임은 작업 호출과는 별도의 작업이므로 역할과 개인을 수동으로 상호 연결해야 합니다.

사용자가 역할을 수임할 때 고유한 역할 세션 이름을 지정하여 이러한 작업을 간단하게 수행할 수 있습니다. 역할을 지정하는 `role_session_name` 파일의 명명된 각 프로파일에 `config` 파라미터를 추가하여 이를 수행합니다. `role_session_name` 값은 `AssumeRole` 작업에 전달되고 역할 세션ARN의 일부가 됩니다. 로그에는 로깅된 모든 작업에 대한 AWS CloudTrail 로그에도 포함됩니다.

예를 들어, 다음과 같이 역할 기반 프로파일을 생성할 수 있습니다.

```
[profile namedsessionrole]
role_arn = arn:aws:iam::234567890123:role/SomeRole
source_profile = default
```

```
role_session_name = Session_Maria_Garcia
```

그러면 역할 세션에 다음 이 포함됩니다ARN.

```
arn:aws:iam::234567890123:assumed-role/SomeRole/Session_Maria_Garcia
```

또한 모든 AWS CloudTrail 로그에는 각 작업에 대해 캡처된 정보에 역할 세션 이름이 포함됩니다.

## 웹 자격 증명을 사용한 역할 수임

웹 ID 페더레이션 및 Open ID Connect(OIDC)를 사용하여 가 역할을 수임 AWS CLI 해야 함을 나타내도록 프로파일을 구성할 수 있습니다. 프로필에서 이를 지정하면 AWS CLI 에서 자동으로 해당 AWS STS `AssumeRoleWithWebIdentity` 호출을 수행합니다.

> **Note**
>
> IAM 역할을 사용하는 프로필을 지정하면 AWS CLI 에서 적절한 호출을 수행하여 임시 보안 인증 정보를 검색합니다. 이러한 자격 증명은 `~/.aws/cli/cache`에 저장됩니다. 동일한 프로파일을 지정하는 후속 AWS CLI 명령은 만료될 때까지 캐시된 임시 자격 증명을 사용합니다. 이때 는 자격 증명을 AWS CLI 자동으로 새로 고칩니다.

웹 자격 증명 연동을 사용하여 임시 자격 증명을 검색하고 사용하려면 공유 프로파일에서 다음 구성 값을 지정할 수 있습니다.

role_arn

수임할 역할ARN의 를 지정합니다.

web_identity_token_file

자격 증명 공급자가 제공하는 OAuth 2.0 액세스 토큰 또는 OpenID Connect ID 토큰이 포함된 파일의 경로를 지정합니다. AWS CLI 에서 이 파일을 로드하고 해당 내용을 `WebIdentityToken` 작업에 대한 `AssumeRoleWithWebIdentity` 인수로 전달합니다.

role_session_name

이 역할 수임 세션에 적용된 선택적 이름을 지정합니다.

다음은 웹 자격 증명 프로파일로 역할 수임을 구성하는 데 필요한 최소 양의 구성에 대한 예입니다.

```
# In ~/.aws/config

[profile web-identity]
role_arn=arn:aws:iam:123456789012:role/RoleNameToAssume
web_identity_token_file=/path/to/a/token
```

환경 변수를 사용하여 이 구성을 제공할 수도 있습니다.

AWS_ROLE_ARN

수임할 역할ARN의 입니다.

AWS_WEB_IDENTITY_TOKEN_FILE

웹 자격 증명 토큰 파일의 경로입니다.

AWS_ROLE_SESSION_NAME

이 역할 수임 세션에 적용된 이름입니다.

> **Note**
>
> 이러한 환경 변수는 현재 웹 자격 증명 공급자의 역할 수임에만 적용됩니다. 일반 역할 수임 공급자 구성에는 적용되지 않습니다.

## 캐시된 자격 증명 지우기

역할을 사용하면 는 임시 보안 인증 정보가 만료될 때까지 로컬에서 AWS CLI 캐시합니다. 다음 번에 이를 사용하려고 할 때 는 사용자를 대신하여 이를 갱신하려고 AWS CLI 시도합니다.

역할의 임시 자격 증명이 취소된 경우에는 해당 자격 증명이 자동으로 갱신되지 않고 사용 시도가 실패합니다. 그러나 캐시를 삭제하여 가 새 보안 인증 정보를 검색하도록 강제 AWS CLI 할 수 있습니다.

Linux 또는 macOS

```
$ rm -r ~/.aws/cli/cache
```

Windows

```
C:\> del /s /q %UserProfile%\.aws\cli\cache
```

# 에 대한 IAM 사용자 자격 증명을 사용하여 인증 AWS CLI

> **Warning**
>
> 보안 위험을 방지하려면 특별히 제작된 소프트웨어를 개발하거나 실제 데이터로 작업할 때 IAM 사용자를 인증에 사용하지 마세요. 대신 AWS IAM Identity Center과 같은 보안 인증 공급자를 통한 페더레이션을 사용하십시오.

이 섹션에서는 IAM 사용자와 함께 기본 설정을 구성하는 방법을 설명합니다. 여기에는 `config` 및 `credentials` 파일을 사용한 보안 보안 인증이 포함됩니다.

주제



## 1단계: IAM 사용자 생성

IAM 사용 설명서의 IAM 사용자 생성(콘솔) 절차에 따라 IAM 사용자를 생성합니다.

- 권한 옵션에서 이 사용자에게 권한을 할당하려는 방법에 대한 정책 직접 연결을 선택합니다.
- 대부분의 "시작하기" SDK 자습서에서는 Amazon S3 서비스를 예로 사용합니다. 애플리케이션에 Amazon S3에 대한 전체 액세스 권한을 제공하려면 이 사용자에게 연결할 `AmazonS3FullAccess` 정책을 선택하세요.

## 2단계: 액세스 키 가져오기

1. 에 로그인 AWS Management Console 하고 에서 IAM 콘솔을 엽니다https://console.aws.amazon.com/iam/.
2. IAM 콘솔의 탐색 창에서 사용자를 선택한 다음 이전에 생성한 사용자의 **User name** 를 선택합니다.

3. 사용자 페이지에서 보안 보안 인증 페이지를 선택합니다. 그런 다음 액세스 키에서 액세스 키 생성을 선택합니다.
4. 액세스 키 생성 1단계에서 명령줄 인터페이스(CLI)를 선택합니다.
5. 액세스 키 만들기 2단계에서 선택적 태그를 입력하고 다음을 선택합니다.
6. 액세스 키 생성 3단계에서 .csv 파일 다운로드를 선택하여 IAM 사용자의 액세스 키와 보안 액세스 키가 있는 `.csv` 파일을 저장합니다. 나중에 이 정보가 필요합니다.
7. 완료(Done)를 선택합니다.

# 구성 AWS CLI

일반적으로 에는 다음과 같은 정보가 AWS CLI 필요합니다.

- 액세스 키 ID
- 보안 액세스 키
- AWS 리전
- 출력 형식

는 이 정보를 `credentials` 파일에 이름이 지정된 프로필(설정 모음)`default`에 AWS CLI 저장합니다. 기본적으로 이 프로필의 정보는 사용할 프로필을 명시적으로 지정하지 않는 AWS CLI 명령을 실행할 때 사용됩니다. `credentials` 파일에 대한 자세한 내용은 의 구성 및 보안 인증 파일 설정 AWS CLI 단원을 참조하세요.

를 구성하려면 다음 절차 중 하나를 AWS CLI사용합니다.

주제

- aws configure 사용하기

## `aws configure` 사용하기

일반적으로 이 `aws configure` 명령은 AWS CLI 설치를 설정하는 가장 빠른 방법입니다. 이 구성 마법사는 시작하는 데 필요한 각 정보를 입력하라는 메시지를 표시합니다. `--profile` 옵션을 사용하여 달리 지정하지 않는 한 는 이 정보를 `default` 프로파일에 AWS CLI 저장합니다.

다음 예에서는 샘플 값을 사용하여 `default` 프로필을 구성합니다. 다음 섹션에 설명된 대로 해당 값을 사용자 고유의 값으로 바꿉니다.

```
$ 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
```

다음 예에서는 샘플 값을 사용하여 userprod로 이름이 지정된 프로필을 구성합니다. 다음 섹션에 설명된 대로 해당 값을 사용자 고유의 값으로 바꿉니다.

```
$ 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
```

# 에서 Amazon EC2 인스턴스 메타데이터를 보안 인증으로 사용 AWS CLI

Amazon Elastic Compute Cloud(AmazonEC2) 인스턴스 AWS CLI 내에서 를 실행할 때 명령에 보안 인증 정보 제공을 간소화할 수 있습니다. 각 Amazon EC2 인스턴스에는 가 AWS CLI 임시 자격 증명을 직접 쿼리할 수 있는 메타데이터가 포함되어 있습니다. IAM 역할이 인스턴스에 연결되면 는 인스턴스 메타데이터에서 자격 증명을 AWS CLI 자동으로 안전하게 검색합니다.

이 서비스를 비활성화하려면 AWS_EC2_METADATA_DISABLED 환경 변수를 사용합니다.

주제

- 사전 조건
- Amazon EC2 메타데이터에 대한 프로필 구성

## 사전 조건

에서 Amazon EC2 자격 증명을 사용하려면 다음을 완료해야 AWS CLI합니다.

- AWS CLI를 설치하고 구성합니다. 자세한 내용은 설치 AWS CLI 및 에 대한 인증 및 액세스 자격 증명 AWS CLI 단원을 참조하세요.
- 구성 파일과 명명된 프로파일을 파악합니다. 자세한 내용은 의 구성 및 보안 인증 파일 설정 AWS CLI 단원을 참조하십시오.

- 필요한 리소스에 액세스할 수 있는 AWS Identity and Access Management (IAM) 역할을 생성하여 시작할 때 Amazon EC2 인스턴스에 연결했습니다. 자세한 내용은 IAM Amazon 사용 설명서의 Amazon 정책 EC2 및 IAM 사용 설명서의 Amazon EC2 인스턴스에서 실행되는 애플리케이션에 AWS 리소스에 대한 액세스 권한 부여를 참조하세요. EC2

## Amazon EC2 메타데이터에 대한 프로필 구성

호스팅 Amazon EC2 인스턴스 프로파일에서 사용할 수 있는 보안 인증 정보를 사용하도록 지정하려면 구성 파일의 명명된 프로파일에서 다음 구문을 사용합니다. 자세한 지침은 다음 단계를 참조하십시오.

```
[profile profilename]
role_arn = arn:aws:iam::123456789012:role/rolename
credential_source = Ec2InstanceMetadata
region = region
```

1. 구성 파일에 프로파일을 생성합니다.

   ```
   [profile profilename]
   ```

2. 필요한 리소스에 액세스할 수 있는 IAM arn 역할을 추가합니다.

   ```
   role_arn = arn:aws:iam::123456789012:role/rolename
   ```

3. 자격 증명 소스로 `Ec2InstanceMetadata`를 지정합니다.

   ```
   credential_source = Ec2InstanceMetadata
   ```

4. 리전을 설정합니다.

   ```
   region = region
   ```

예

다음 예제에서는 *marketingadminrole* 역할 및 는 라는 Amazon EC2 인스턴스 프로파일에서 *us-west-2* 리전을 사용합니다*marketingadmin*.

```
[profile marketingadmin]
role_arn = arn:aws:iam::123456789012:role/marketingadminrole
```

```
credential_source = Ec2InstanceMetadata
region = us-west-2
```

# 에서 외부 프로세스를 사용하여 보안 인증 정보 소싱 AWS CLI

> **Warning**
>
> 이 주제에서는 외부 프로세스에서 자격 증명을 소싱하는 방법을 알아봅니다. 자격 증명을 생성하는 명령이 승인되지 않은 프로세스나 사용자가 액세스할 수 있게 된 경우에는 이것이 보안 위험이 될 수 있습니다. 자격 증명 손상 위험을 줄 AWS 이려면 AWS CLI 및 에서 제공하는 지원되는 안전한 대안을 사용하는 것이 좋습니다. config 파일과 유출 방지를 도와주는 모든 파일 및 도구를 보호하고 있는지 확인합니다.
>
> SDKs 사용자 지정 보안 인증 도구는 및 가 이러한 정보를 캡처하고 기록할 AWS CLI 수 있어 권한이 없는 사용자에게 노출될 수 StdErr 있으므로 에 보안 정보를 쓰지 않도록 해야 합니다.

에서 직접 지원하지 않는 보안 인증 정보를 생성하거나 조회하는 메서드가 있는 AWS CLI경우 config 파일에서 credential_process 설정을 구성하여 가 이를 AWS CLI 사용하도록 구성할 수 있습니다.

예를 들어 config 파일에 다음과 유사한 항목을 포함시킬 수 있습니다.

```
[profile developer]
credential_process = /opt/bin/awscreds-custom --username helen
```

구문

기존 운영 체제와 호환되는 방식으로 이 문자열을 생성하려면 다음 규칙을 따르십시오.

- 경로 또는 파일 이름에 공백이 있으면 전체 경로와 파일 이름을 큰 따옴표(" ")로 묶습니다. 경로 및 파일 이름은 다음 문자만 포함할 수 있습니다. A-Z a-z 0-9 - _ . 공백
- 파라미터 이름이나 파라미터 값에 공백이 있으면 해당 요소를 큰 따옴표(" ")로 묶습니다. 전체 페어가 아니라 이름 또는 값만 묶으십시오.
- 문자열 안에 환경 변수를 포함하지 마십시오. 예를 들어 $HOME 또는 %USERPROFILE%를 포함할 수 없습니다.
- 홈 폴더를 ~로 지정하지 마십시오. 전체 경로를 지정해야 합니다.

Windows용 예

```
credential_process = "C:\Path\To\credentials.cmd" parameterWithoutSpaces "parameter
 with spaces"
```

Linux 또는 macOS용 예

```
credential_process = "/Users/Dave/path/to/credentials.sh" parameterWithoutSpaces
 "parameter with spaces"
```

자격 증명 프로그램에서 예상되는 출력

는 프로파일에 지정된 대로 명령을 AWS CLI 실행한 다음 에서 데이터를 읽습니다STDOUT. 지정하는 명령은 다음 구문과 STDOUT 일치하는 JSON 출력을 에 생성해야 합니다.

```
{
  "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**
>
> 이 문서의 작성일 현재, `Version` 키는 1로 설정되어 있습니다. 구조가 발전하면서 시간에 따라 이 값이 증가할 수 있습니다.

`Expiration` 키는 ISO8601 형식의 타임스탬프입니다. `Expiration` 키가 도구의 출력에 없는 경우는 보안 인증이 새로 고쳐지지 않는 장기 보안 인증 정보라고 CLI 가정합니다. 그렇지 않은 경우 자격 증명은 임시 자격 증명으로 간주되며, 기간이 만료되기 전에 `credential_process` 명령을 다시 실행하면 자동으로 새로 고침됩니다.

> **Note**
>
> AWS CLI 는 외부 프로세스 자격 증명을 수임-역할 자격 증명을 수행하는 방식으로 캐시하지 않습니다. 캐싱이 필요한 경우에는 외부 프로세스에서 이를 실행해야 합니다.

외부 프로세스는 자격 증명을 검색하는 동안 오류가 발생했음을 나타내기 위해 0이 아닌 반환 코드를 반환할 수 있습니다.

# 사용 AWS CLI

이 섹션에서는 구성 the section called “엔드포인트” 섹션에서 다루는 세부 정보를 넘어 AWS Command Line Interface (AWS CLI)에서 사용할 수 있는 일반적인 사용, 일반적인 기능 및 옵션에 대한 포괄적인 개요를 제공합니다.

이 가이드에서는 기본 구조, 형식 지정 및 필터링 기능을 포함하여 쓰기 AWS CLI 명령의 기본 측면을 자세히 살펴봅니다. 이러한 핵심 요소를 이해하면 복잡한 웹 기반 콘솔을 탐색할 필요 없이 필요한 리소스와 작업을 정확하게 대상으로 하는 명령을 구성할 수 있습니다.

또한 에 사용할 수 있는 도움말 콘텐츠와 설명서도 강조 표시됩니다 AWS CLI. 기본 제공 명령줄 도움말에서 포괄적인 AWS CLI 참조 가이드 의 기능을 탐색하는 데 도움이 되는 정보에 액세스할 수 있습니다 AWS CLI.

AWS 서비스 특정 예제 및 사용 사례는 코드 예시 또는 AWS CLI 참조 가이드 를 참조하세요. 이는 명령별 정보를 제공하고 다양한 에 대해 를 활용하는 방법에 AWS CLI 대한 예를 보여줍니다 AWS 서비스.

> **Note**
>
> 기본적으로 는 TCP 포트 443HTTPS에서 를 사용하여 AWS 서비스 에 요청을 AWS CLI 보냅니다. 를 성공적으로 사용하려면 이 포트에서 아웃바운드 연결을 수행할 수 있어야 AWS CLI합니다.

이 안내서의 주제

- 에 대한 도움말 및 리소스 액세스 AWS CLI
- 의 명령 구조 AWS CLI
- 에서 파라미터 값 지정 AWS CLI
- 에서 명령 출력 제어 AWS CLI
- 의 명령줄 반환 코드 AWS CLI
- 에서 별칭 생성 및 사용 AWS CLI

# 에 대한 도움말 및 리소스 액세스 AWS CLI

이 주제에서는 AWS Command Line Interface ()에 대한 도움말 콘텐츠에 액세스하는 방법을 설명합니다AWS CLI.

주제



## 기본 제공 AWS CLI help 명령

AWS Command Line Interface ()를 사용할 때 모든 명령에 대한 도움말을 얻을 수 있습니다AWS CLI. 이를 위해 명령 이름 끝에 `help`를 입력하기만 하면 됩니다.

예를 들어 다음 명령은 일반 AWS CLI 옵션 및 사용 가능한 최상위 명령에 대한 도움말을 표시합니다.

```
$ aws help
```

다음 명령은 사용 가능한 Amazon Elastic Compute Cloud(AmazonEC2)별 명령을 표시합니다.

```
$ aws ec2 help
```

다음 예제에서는 Amazon EC2 `DescribeInstances` 작업에 대한 자세한 도움말을 보여줍니다. 도움말에는 입력 파라미터, 사용 가능한 필터, 출력으로 포함되는 항목에 대한 설명이 포함됩니다. 해당 명령의 일반적인 변형을 입력하는 방법을 보여주는 예제도 포함되어 있습니다.

```
$ aws ec2 describe-instances help
```

각 명령에 대한 도움말은 다음과 같은 6개 섹션으로 나뉩니다.

이름

명령의 이름입니다.

```
NAME
```

```
       describe-instances -
```

### 설명

명령이 호출하는 API 작업에 대한 설명입니다.

```
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.
...
```

### 시놉시스

명령 및 옵션 사용을 위한 기본 구문입니다. 옵션을 대괄호로 표시할 경우 선택적인 옵션이거나, 기본값을 가지거나, 사용할 수 있는 대체 옵션을 갖고 있습니다.

```
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]
```

예를 들어 `describe-instances`에는 현재 계정 및 AWS 리전의 모든 인스턴스를 설명하는 기본 동작이 있습니다. 1개 이상의 인스턴스를 설명할 `instance-ids` 목록을 선택적으로 지정할 수 있습니다. `dry-run`은 값을 가지지 않는 선택 사항인 부울 플래그입니다. 부울 플래그를 사용하려면 표시된 값을 지정합니다. 이 경우 `--dry-run` 또는 `--no-dry-run`입니다. 마찬가지로 `--generate-cli-skeleton`도 값을 갖고 있지 않습니다. 옵션 사용 시 조건이 있는 경우 해당 조건을 OPTIONS 섹션에서 설명하거나 예제에 표시합니다.

### 옵션

개요에 표시된 각 옵션에 대한 설명입니다.

```
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  DryRun-
          Operation . Otherwise, it is UnauthorizedOperation .

       --instance-ids (list)
          One or more instance IDs.

          Default: Describes all your instances.
...
```

예제:

명령 및 해당 옵션의 사용을 보여 주는 예제입니다. 필요한 명령 또는 사용 사례에 사용할 수 있는 예제가 없는 경우 이 페이지의 피드백 링크 또는 AWS CLI 명령에 대한 도움말 페이지의 명령 참조를 사용하여 요청하세요.

```
    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"
...
```

출력

의 응답에 포함되는 각 필드 및 데이터 형식에 대한 설명입니다 AWS

`describe-instances`의 경우 출력은 예약 객체의 목록이며, 각 객체에는 연관된 인스턴스에 대한 정보를 포함하는 여러 필드 및 객체가 포함됩니다. 이 정보는 Amazon 에서 사용하는 API 예약 데이터 유형에 대한 설명서에서 제공됩니다EC2.

```
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.
...
```

가 출력을 로 AWS CLI 렌더링하면 다음 예제와 마찬가지로 예약 객체 배열JSON이 됩니다.

```
{
    "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"
                    },
...
```

각 예약 객체에는 예약을 설명하는 필드와 인스턴스 객체의 배열이 포함됩니다. 각 배열에는 고유한 필드(예: `PublicDnsName`) 및 이를 설명하는 객체(예: `State`)가 있습니다.

> **Windows 사용자**
>
> 도움말 파일을 한 번에 한 페이지씩 보기 위해 help 명령의 출력을 `more` 명령에 파이프 (|) 할 수 있습니다. 스페이스바 또는 를 눌러 문서를 더 PgDn 많이 보고 종료**q**합니다.
>
> ```
> C:\> aws ec2 describe-instances help | more
> ```

# AWS CLI 참조 가이드

도움말 파일에는 명령줄에서 탐색하거나 볼 수 없는 링크가 포함됩니다. 온라인 AWS CLI 버전 1 참조 가이드 버전 . 참조에는 모든 AWS CLI 명령에 대한 도움말 콘텐츠도 포함되어 있습니다. 이 설명은 쉽게 탐색하고 모바일, 태블릿 또는 데스크톱 화면에서 볼 수 있도록 제공됩니다.

# API 설명서

의 모든 명령은 AWS 서비스의 퍼블릭 에 대한 요청에 AWS CLI 해당합니다API. 퍼블릭이 있는 각 서비스API에는 AWS 설명서 웹 사이트의 서비스 홈 페이지에서 찾을 수 있는 API 참조가 있습니다. API 참조의 내용은 API 의 구성 방식과 사용되는 프로토콜에 따라 달라집니다. 일반적으로 API 참조에는 에서 지원하는 작업API, 서비스에서 주고받는 데이터, 서비스가 보고할 수 있는 오류 조건에 대한 자세한 정보가 포함되어 있습니다.

API 문서 섹션

- 작업 - 각 작업과 해당 파라미터에 대한 세부 정보(길이 또는 콘텐츠에 대한 제약, 기본값 등)입니다. 이 작업에 발생할 수 있는 오류를 나열합니다. 각 작업은 의 하위 명령에 해당합니다 AWS CLI.
- 데이터 유형 - 명령이 파라미터로 요구하거나 요청에 대한 응답으로 반환할 수 있는 구조에 대한 자세한 정보입니다.
- 범용 파라미터 - 서비스의 모든 작업에서 공유하는 파라미터에 대한 세부 정보입니다.
- 범용 오류 - 모든 서비스 작업에서 반환할 수 있는 오류에 대한 세부 정보입니다.

각 섹션의 이름 및 가용성은 서비스에 따라 다를 수 있습니다.

> **서비스별 CLIs**
>
> 일부 서비스에는 모든 서비스에서 작동하도록 단일 가 생성되기 전에CLI는 와 별도의 날짜가 AWS CLI 있습니다. 이러한 서비스별 설명서에는 서비스의 설명서 페이지와 연결된 별도의 설명서가 CLIs 있습니다. 서비스별 설명서CLIs는 에 적용되지 않습니다 AWS CLI.

# 오류 해결

AWS CLI 오류 진단 및 수정에 대한 도움말은 섹션을 참조하세요오류 해결.

## 추가 도움말

AWS CLI 문제에 대한 추가 도움이 필요하면 의 AWS CLI 커뮤니티를 참조하세요GitHub.

# 의 명령 구조 AWS CLI

이 주제에서는 AWS Command Line Interface (AWS CLI) 명령이 구조화되는 방법과 대기 명령을 사용하는 방법을 다룹니다.

주제

- 명령 구조
- wait 명령

## 명령 구조

는 명령줄에서 다음 순서로 지정해야 하는 멀티파트 구조를 AWS CLI 사용합니다.

1. aws 프로그램에 대한 기본 호출.
2. 일반적으로 에서 지원하는 AWS 서비스에 해당하는 최상위 명령입니다 AWS CLI.
3. 어떤 작업을 수행할지 지정하는 하위 명령 입니다.
4. 작업에 필요한 일반 AWS CLI 옵션 또는 파라미터입니다. 처음 세 개 파트를 따르기만 하면 어떤 순서로든 지정할 수 있습니다. 독점적인 파라미터를 여러 번 지정하면 마지막 값만 적용됩니다.

```
$ aws <command> <subcommand> [options and parameters]
```

파라미터는 숫자, 문자열, 목록, 맵 및 JSON 구조와 같은 다양한 유형의 입력 값을 취할 수 있습니다. 무엇이 지원되는지는 지정하는 명령 및 하위 명령에 따라 달라집니다.

예시

Amazon S3

다음 예제에서는 모든 Amazon S3 버킷을 나열합니다.

```
$ aws s3 ls
2018-12-11 17:08:50 amzn-s3-demo-bucket1
2018-12-14 14:55:44 amzn-s3-demo-bucket2
```

Amazon S3 명령에 대한 자세한 내용은 AWS CLI 명령 참조에서 `aws s3` 단원을 참조하세요.

AWS CloudFormation

다음 `create-change-set`명령 예제는 cloudformation 스택 이름을 로 변경합니다.*my-change-set*.

```
$ aws cloudformation create-change-set --stack-name my-stack --change-set-name my-change-set
```

AWS CloudFormation 명령에 대한 자세한 내용은 명령 참조`aws cloudformation`의 섹션을 참조하세요. AWS CLI

## wait 명령

일부 AWS 서비스에는 `wait` 명령을 사용할 수 있습니다. `aws wait`를 사용하는 명령은 일반적으로 명령이 완료될 때까지 기다린 후 다음 단계로 넘어갑니다. 이는 wait 명령이 실패할 경우 후속 단계로 이동하지 않도록 wait 명령을 사용할 수 있으므로 멀티파트 명령 또는 스크립팅에 특히 유용합니다.

는 명령줄에서 다음 순서로 지정해야 하는 멀티파트 구조를 `wait` 명령에 AWS CLI 사용합니다.

1. `aws` 프로그램에 대한 기본 호출.
2. 일반적으로 에서 지원하는 AWS 서비스에 해당하는 최상위 명령입니다 AWS CLI.
3. `wait` 명령.
4. 어떤 작업을 수행할지 지정하는 하위 명령 입니다.
5. 작업에 필요한 일반 CLI 옵션 또는 파라미터입니다. 처음 세 개 파트를 따르기만 하면 어떤 순서로든 지정할 수 있습니다. 독점적인 파라미터를 여러 번 지정하면 마지막 값만 적용됩니다.

```
$ aws <command> wait <subcommand> [options and parameters]
```

파라미터는 숫자, 문자열, 목록, 맵 및 JSON 구조와 같은 다양한 유형의 입력 값을 취할 수 있습니다. 무엇이 지원되는지는 지정하는 명령 및 하위 명령에 따라 달라집니다.

> **Note**
>
> 모든 AWS 서비스가 `wait` 명령을 지원하는 것은 아닙니다. AWS CLI 참조 가이드 참조하여 서비스가 `wait` 명령을 지원하는지 확인하세요.

예시

AWS CloudFormation

다음 wait change-set-create-complete 명령 예제는 *my-change-set* 에서 설정 변경 *my-stack* 스택을 실행할 준비가 되었습니다.

```
$ aws cloudformation wait change-set-create-complete --stack-name my-stack --change-set-name my-change-set
```

AWS CloudFormation wait 명령에 대한 자세한 내용은 AWS CLI 명령 참조의 wait 단원을 참조하세요.

AWS CodeDeploy

다음 wait deployment-successful 명령 예제는 *d-A1B2C3111* 배포가 성공적으로 완료되었습니다.

```
$ aws deploy wait deployment-successful --deployment-id d-A1B2C3111
```

AWS CodeDeploy wait 명령에 대한 자세한 내용은 AWS CLI 명령 참조의 wait 단원을 참조하세요.

# 에서 파라미터 값 지정 AWS CLI

AWS Command Line Interface (AWS CLI)에 사용되는 많은 파라미터는 다음 aws ec2 create-key-pair 명령 예제의 키 페어 이름과 같은 간단한 문자열 또는 숫자 값my-key-pair입니다.

```
$ aws ec2 create-key-pair --key-name my-key-pair
```

명령의 형식은 터미널마다 다를 수 있습니다. 예를 들어, 대부분의 터미널은 대소문자를 구분하지만 PowerShell은 대소문자를 구분하지 않습니다. 즉, 다음 두 명령 예시는 대소문자를 구분하는 터미널에서 서로 다른 결과를 얻습니다. MyFile*.txt와 myfile*.txt가 서로 다른 파라미터로 간주되기 때문입니다.

그러나 PowerShell 는 이러한 요청을 표시된 것과 동일한 MyFile*.txt 파라미터와 myfile*.txt 동일하게 처리할 것입니다. 다음 명령 예제는 aws s3 cp 명령을 사용하여 이러한 파라미터들을 보여줍니다.

```
$ aws s3 cp . s3://amzn-s3-demo-bucket/path --include "MyFile*.txt"
$ aws s3 cp . s3://amzn-s3-demo-bucket/path --include "myfile*.txt"
```

PowerShell의 대/소문자 비민감성에 대한 자세한 내용은 PowerShell 설명서의 about_Case-Sensitivity를 참조하세요.

특수 문자나 공백 문자가 포함된 문자열을 따옴표 또는 리터럴로 묶어야 하는 경우가 있습니다. 이 형식에 대한 규칙도 터미널마다 다를 수 있습니다. 복잡한 파라미터를 따옴표로 묶는 방법에 대한 자세한 내용은 에서 문자열과 함께 따옴표 및 리터럴 사용 AWS CLI 단원을 참조하십시오.

이 주제에서는 가장 일반적인 터미널 형식 지정 규칙을 다룹니다. 터미널에서 파라미터 값을 인식하는 데 문제가 있는 경우 이 섹션의 주제를 검토하고 터미널의 설명서에서 특정 구문 규칙을 확인해야 합니다.

매개 변수 주제

- 의 공통 파라미터 유형 AWS CLI
- 에서 문자열과 함께 따옴표 및 리터럴 사용 AWS CLI
- 의 파일에서 파라미터 로드 AWS CLI
- AWS CLI 의 스켈레톤 및 입력 파일 AWS CLI
- 에서 간이 구문 사용 AWS CLI

## 의 공통 파라미터 유형 AWS CLI

이 단원에서는 몇 가지 공통 파라미터 유형과 일반적으로 필요한 형식에 대해 설명합니다.

특정 명령에 대한 파라미터의 형식 지정에 문제가 있는 경우, 명령 이름 다음에 **help**를 입력하여 도움말을 검토합니다. 각 하위 명령에 대한 도움말에는 옵션의 이름과 설명이 포함되어 있습니다. 옵션의 파라미터 유형이 괄호 안에 나열됩니다. 도움말 보기에 대한 자세한 내용은 the section called “도움받기” 단원을 참조하세요.

파라미터 유형에는 다음이 포함됩니다.

- String
- Timestamp
- 나열
- 불
- Integer
- 이진/blob(이진 대용량 객체) 및 스트리밍 blob
- 맵
- 문서

### String

문자열 파라미터에는 문자 세트의 영숫자 문자, 기호 및 공백이 포함될 수 ASCII 있습니다. 공백이 포함된 문자열은 인용 부호로 묶어야 합니다. 표준 공백 문자 이외의 기호 또는 공백은 사용하지 않고 예기치 않은 결과를 방지하기 위해 터미널의 인용 규칙을 준수하는 것이 좋습니다.

일부 문자열 파라미터는 파일의 이진 데이터를 허용할 수 있습니다. 예제는 이진 파일 단원을 참조하십시오.

### Timestamp

타임스탬프는 ISO 8601 표준에 따라 형식이 지정됩니다. 흔히 'DateTime' 또는 'Date' 파라미터라고 합니다.

```
$ aws ec2 describe-spot-price-history --start-time 2014-10-13T19:00:00Z
```

허용 가능한 형식은 다음과 같습니다.

- *YYYY-MM-DD*T*hh*:*mm*:*ss.sssTZD (UTC)*, 예: 2014-10-01T20:30:00.000Z
- *YYYY-MM-DD*T*hh*:*mm*:*ss.sssTZD (with offset)*, 예: 2014-10-01T12:30:00.000-08:00
- *YYYY-MM-DD*, 예: 2014-10-01
- 초 단위의 Unix 시간, 예: 1412195400 이를 Unix Epoch 시간이라고 하며 1970년 1월 1일 자정 이후 초 수를 나타냅니다UTC.

`cli_timestamp_format` 파일 설정을 사용하여 타임스탬프 형식을 설정할 수 있습니다.

### 나열

공백으로 구분된 하나 이상의 문자열입니다. 문자열 항목에 공백이 포함되어 있으면 해당 항목 앞뒤에 인용 부호를 사용해야 합니다. 예기치 않은 결과를 방지하기 위해 터미널의 인용 규칙을 준수합니다.

```
$ aws ec2 describe-spot-price-history --instance-types m1.xlarge m1.medium
```

### 불

옵션을 켜거나 끄는 이진 플래그입니다. 예를 들어, `ec2 describe-spot-price-history`에는 지정할 경우 실제 쿼리는 실행하지 않고 서비스에 대해 쿼리를 검증하는 부울 `--dry-run` 파라미터가 있습니다.

```
$ aws ec2 describe-spot-price-history --dry-run
```

출력은 명령이 제대로 구성되었는지 여부를 나타냅니다. 또한 이 명령에는 명령을 정상적으로 실행해야 함을 명시적으로 표시하는 데 사용할 수 있는 `--no-dry-run` 버전의 파라미터도 포함됩니다. 기본 동작이기 때문에 반드시 포함할 필요는 없습니다.

### Integer

부호가 없는 정수입니다.

```
$ aws ec2 describe-spot-price-history --max-items 5
```

### 이진/blob(이진 대용량 객체) 및 스트리밍 blob

에서 이진 값을 명령줄에 직접 문자열로 전달할 AWS CLI수 있습니다. 다음과 같은 두 가지 유형의 Blob이 있습니다.

- Blob
- 스트리밍 Blob

#### Blob

`blob` 유형의 파라미터에 값을 전달하려면 `fileb://` 접두사를 사용하여 이진 데이터가 포함된 로컬 파일의 경로를 지정해야 합니다. `fileb://` 접두사를 사용하여 참조된 파일은 항상 인코딩되지 않은 원시 이진 값으로 처리됩니다. 지정된 경로는 현재 작업 디렉터리를 기준으로 하는 것으로 해석됩니다. 예를 들어 `--plaintext`의 `aws kms encrypt` 파라미터는 BLOB입니다.

```
$ aws kms encrypt \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --plaintext fileb://ExamplePlaintextFile \
    --output text \
    --query CiphertextBlob | base64 \
    --decode > ExampleEncryptedFile
```

#### 스트리밍 Blob

`aws cloudsearchdomain upload-documents` 등의 스트리밍 Blob은 접두사를 사용하지 않습니다. 대신 스트리밍 blob 파라미터는 직접 파일 경로를 사용하여 형식이 지정됩니다. 다음 예제

에서는 `aws cloudsearchdomain upload-documents` 명령에 직접 파일 경로 `document-batch.json`을 사용합니다.

```
$ aws cloudsearchdomain upload-documents \
    --endpoint-url https://doc-my-domain.us-west-1.cloudsearch.amazonaws.com \
    --content-type application/json \
    --documents document-batch.json
```

### 맵

JSON 또는 에서 의 단축형 구문을 사용하여 지정된 키-값 페어 세트CLI입니다. 다음 JSON 예제는 맵 파라미터 를 사용하여 my-table이라는 Amazon DynamoDB 테이블에서 항목을 읽습니다`--key`. 파라미터는 중첩 JSON 구조에서 숫자 값이 1인 id라는 기본 키를 지정합니다.

명령줄에서 더 고급으로 JSON 사용하려면 와 같은 명령줄 JSON 프로세서를 사용하여 JSON 문자열을 생성하는 `jq`것이 좋습니다. 에 대한 자세한 내용은 의 jq 리포지토리를 `jq`참조하세요GitHub.

```
$ aws dynamodb get-item --table-name my-table --key '{"id": {"N":"1"}}'

{
    "Item": {
        "name": {
            "S": "John"
        },
        "id": {
            "N": "1"
        }
    }
}
```

### 문서

> **Note**
>
> 약식 구문은 문서 유형과 호환되지 않습니다.

문서 유형은 문자열 JSON 내에 임베드할 필요 없이 데이터를 전송하는 데 사용됩니다. 문서 유형을 사용하면 서비스에서 보다 유연한 데이터 유형을 사용할 수 있도록 임의의 스키마를 제공할 수 있습니다.

이렇게 하면 값을 이스케이프할 필요 없이 JSON 데이터를 전송할 수 있습니다. 예를 들어 다음과 같은 이스케이프된 JSON 입력을 사용하는 대신

```
{"document": "{\"key\":true}"}
```

다음 문서 유형을 사용할 수 있습니다.

```
{"document": {"key": true}}
```

문서 유형에 유효한 값

문서 유형의 유연한 특성으로 인해 유효한 값 유형이 여러 개 있습니다. 유효한 값은 다음과 같습니다.

String

```
--option '"value"'
```

숫자

```
--option 123
--option 123.456
```

불

```
--option true
```

Null

```
--option null
```

배열

```
--option '["value1", "value2", "value3"]'
--option '["value", 1, true, null, ["key1", 2.34], {"key2": "value2"}]'
```

객체

```
--option '{"key": "value"}'
--option '{"key1": "value1", "key2": 123, "key3": true, "key4": null, "key5":
 ["value3", "value4"], "key6": {"value5": "value6"}'
```

# 에서 문자열과 함께 따옴표 및 리터럴 사용 AWS CLI

AWS CLI에는 주로 두 가지 방법으로 작은따옴표와 큰따옴표가 사용됩니다.



## 공백이 포함된 문자열 주위에 따옴표 사용

파라미터 이름과 그 값은 명령줄에서 공백으로 구분됩니다. 문자열 값에 포함된 공백이 있는 경우 가 해당 값과 다음 파라미터 이름 간의 칸막이로 공백을 잘못 해석 AWS CLI 하지 않도록 전체 문자열을 따옴표로 묶어야 합니다. 사용하는 따옴표 유형은 실행 중인 운영 체제 AWS CLI 에 따라 다릅니다.

Linux and macOS

작은따옴표(' ')를 사용합니다.

```
$ aws ec2 create-key-pair --key-name 'my key pair'
```

따옴표 사용에 대한 자세한 내용은 권장 셸에 대한 사용 설명서를 참조하십시오.

PowerShell

작은따옴표(권장)

작은따옴표 ' '는 `verbatim` 문자열이라고 합니다. 문자열은 입력한 대로 명령으로 전달되므로 PowerShell 변수가 전달되지 않습니다.

```
PS C:\> aws ec2 create-key-pair --key-name 'my key pair'
```

큰따옴표

큰따옴표 " "는 `expandable` 문자열이라고 합니다. 변수는 확장 가능한 문자열로 전달될 수 있습니다.

```
PS C:\> aws ec2 create-key-pair --key-name "my key pair"
```

견적 사용에 대한 자세한 내용은 Microsoft 문서 의 견적 규칙 정보를 참조하세요. PowerShell

Windows command prompt

큰따옴표(" ")를 사용합니다.

```
C:\> aws ec2 create-key-pair --key-name "my key pair"
```

선택적으로, 공백 대신에 등호(=)를 사용하여 파라미터 이름을 값에서 분리할 수 있습니다. 일반적으로 파라미터 값이 하이픈으로 시작되는 경우에만 필요합니다.

```
$ aws ec2 delete-key-pair --key-name=-mykey
```

### 문자열 안에 따옴표 사용

문자열에는 따옴표가 포함될 수 있으며 셸이 제대로 작동하려면 따옴표를 이스케이프 처리해야 할 수 있습니다. 일반적인 파라미터 값 유형 중 하나는 JSON 문자열입니다. 여기에는 JSON 구조의 " " 각 요소 이름 및 값에 공백과 큰따옴표가 포함되어 있기 때문에 복잡합니다. 명령줄에 JSON형식이 지정된 파라미터를 입력하는 방법은 운영 체제에 따라 다릅니다.

명령줄에서 더 고급JSON으로 사용하려면 와 같은 명령줄 JSON 프로세서를 사용하여 JSON 문자열을 생성하는 jq것이 좋습니다. 에 대한 자세한 내용은 의 jq 리포지토리를 jq참조하세요GitHub.

Linux and macOS

Linux 및 macOS에서 문자열을 해석하려면 다음 예제와 같이 문자 그대로 작은 따옴표를 사용하여 JSON 데이터 구조를 ' '묶습니다. 문자열이 문자 그대로 처리되므로 JSON 문자열에 포함된 큰따옴표를 이스케이프할 필요가 없습니다. JSON 는 작은따옴표로 묶여 있으므로 문자열의 모든 작은따옴표는 이스케이프되어야 하며, 이는 일반적으로 작은따옴표 앞에 백슬래시를 사용하여 수행됩니다\'.

```
$ aws ec2 run-instances \
    --image-id ami-12345678 \
    --block-device-mappings '[{"DeviceName":"/dev/sdb","Ebs":
{"VolumeSize":20,"DeleteOnTermination":false,"VolumeType":"standard"}}]'
```

따옴표 사용에 대한 자세한 내용은 권장 셸에 대한 사용 설명서를 참조하십시오.

PowerShell

작은따옴표(' ') 또는 큰따옴표(" ")를 사용합니다.

작은따옴표(권장)

작은따옴표 ' '는 verbatim 문자열이라고 합니다. 문자열은 입력한 대로 명령으로 전달되므로 PowerShell 변수가 전달되지 않습니다.

JSON 데이터 구조에는 큰따옴표가 포함되어 있으므로 작은따옴표' '로 묶는 것이 좋습니다. 작은따옴표를 사용하는 경우 JSON 문자열에 포함된 큰따옴표를 이스케이프할 필요가 없습니다. 그러나 JSON 구조 ` 내에 백틱을 사용하여 각 단일 따옴표에서 벗어나야 합니다.

```
PS C:\> aws ec2 run-instances `
    --image-id ami-12345678 `
    --block-device-mappings '[{"DeviceName":"/dev/sdb","Ebs":
{"VolumeSize":20,"DeleteOnTermination":false,"VolumeType":"standard"}}]'
```

큰따옴표

큰따옴표" "는 expandable 문자열이라고 합니다. 변수는 확장 가능한 문자열로 전달될 수 있습니다.

큰따옴표를 사용하는 경우 JSON 문자열에 포함된 작은따옴표를 이스케이프할 필요가 없습니다. 그러나 다음 예제와 같이 JSON 구조 ` 내에서 백틱으로 각 큰따옴표를 이스케이프해야 합니다.

```
PS C:\> aws ec2 run-instances `
    --image-id ami-12345678 `
    --block-device-mappings "[{`"DeviceName`":`"/dev/sdb`",`"Ebs`":
{`"VolumeSize`":20,`"DeleteOnTermination`":false,`"VolumeType`":`"standard`"}}]"
```

견적 사용에 대한 자세한 내용은 Microsoft 문서 의 견적 규칙 정보를 참조하세요. PowerShell

> **Warning**
>
> 가 명령을 에 PowerShell 전송하기 전에 일반적인 규칙 PowerShell 또는 CommandLineToArgvW 인용 규칙을 사용하여 명령을 해석할지 여부를 AWS CLI결정합니다. 를 사용하여 를 처리할 때는 PowerShell 백슬래시 로 문자를 이스케이프해야 CommandLineToArgvW합니다\.
> CommandLineToArgvW 의 에 대한 자세한 내용은 Microsoft DevBlogs 에서 CommandLineToArgvW로 따옴표와 백슬래시를 이상하게 처리하면 어떻게 되는지, Microsoft Docs 블로그에서 모든 사람이 명령줄 인수를 잘못된 방식으로 따옴표와 백슬래시 처리, Microsoft Docs에서 CommandLineToArgvW 함수를 PowerShell참조하세요.
> 작은따옴표
> 작은따옴표' '는 verbatim 문자열이라고 합니다. 문자열은 입력한 대로 명령으로 전달되므로 PowerShell 변수가 전달되지 않습니다. 백슬래시(\)로 문자를 이스케이프 처리합니다.
>
> ```
> PS C:\> aws ec2 run-instances `
> ```

```
    --image-id ami-12345678 `
    --block-device-mappings '[{\"DeviceName\":\"/dev/sdb\",\"Ebs\":
{\"VolumeSize\":20,\"DeleteOnTermination\":false,\"VolumeType\":\"standard\"}}]'
```

큰따옴표
큰따옴표 " "는 expandable 문자열이라고 합니다. 변수는 expandable 문자열로 전달될 수 있습니다. 큰따옴표 문자열의 경우 를 사용하여 두 번 이스케이프해야 합니다. `\ 백틱만 사용하는 대신 각 따옴표에 대해. 백틱은 백슬래시를 이스케이프하며, 백슬래시는 CommandLineToArgvW 프로세스의 이스케이프 문자로 사용됩니다.

```
PS C:\> aws ec2 run-instances `
    --image-id ami-12345678 `
    --block-device-mappings "[{`\"DeviceName`\":`\"/dev/sdb`\",`\"Ebs`\":
{`\"VolumeSize`\":20,`\"DeleteOnTermination`\":false,`\"VolumeType`\":`
\"standard`\"}}]"
```

Blob(권장)
JSON 데이터 입력에 대한 PowerShell 인용 규칙을 우회하려면 Blobs를 사용하여 JSON 데이터를 에 직접 전달합니다 AWS CLI. Blob에 대한 자세한 내용은 Blob 섹션을 참조하세요.

Windows command prompt

Windows 명령 프롬프트는 JSON 데이터 구조를 둘러싸기 " " 위해 큰따옴표가 필요합니다. 또한 명령 프로세서가 에 포함된 큰따옴표를 잘못 해석하지 못하도록 하려면 다음 예제와 같이 JSON 데이터 구조 자체 " 내에서 각 큰따옴표를 이스케이프(이전에는 백슬래시 \ 문자 사용)JSON해야 합니다.

```
C:\> aws ec2 run-instances ^
    --image-id ami-12345678 ^
    --block-device-mappings "[{\"DeviceName\":\"/dev/sdb\",\"Ebs\":
{\"VolumeSize\":20,\"DeleteOnTermination\":false,\"VolumeType\":\"standard\"}}]"
```

가장 바깥쪽 큰따옴표만 이스케이프되지 않습니다.

## 의 파일에서 파라미터 로드 AWS CLI

일부 파라미터는 파일 이름을 인수로 예상하며, 이 인수에서 데이터를 AWS CLI 로드합니다. 다른 어떤 파라미터는 파라미터 값을 명령줄에 입력된 텍스트 또는 파일에서 읽은 텍스트로 지정할

수 있습니다. 파일이 필요하든 선택 사항이든 가 파일을 이해할 AWS CLI 수 있도록 파일을 올바르게 인코딩해야 합니다. 파일의 인코딩은 읽는 시스템의 기본 로캘과 일치해야 합니다. Python `locale.getpreferredencoding()` 메서드를 사용하여 이를 확인할 수 있습니다.

> **Note**
>
> 기본적으로 Windows는 텍스트를 UTF-16으로 PowerShell 출력하며, 이는 JSON 파일 및 많은 Linux 시스템에서 사용되는 UTF-8 인코딩과 충돌합니다. 가 결과 파일을 읽을 AWS CLI 수 있도록 명령과 `-Encoding ascii` 함께 PowerShell `Out-File` 를 사용하는 것이 좋습니다.

주제



## 파일에서 파라미터를 로드하는 방법

파라미터가 복잡한 JSON 문자열인 경우와 같이 모든 파라미터를 명령줄 파라미터 값으로 입력하는 대신 파일에서 파라미터 값을 로드하는 것이 편리할 때도 있습니다. 값이 포함된 파일을 지정하려면 URL 다음 형식으로 파일을 지정합니다.

```
file://complete/path/to/file
```

- 처음 두 개의 슬래시 '/' 문자는 사양의 일부입니다. 필수 경로가 '/'로 시작하면 결과는 슬래시 문자 세 개(`file:///folder/file`)입니다.
- 는 실제 파라미터 콘텐츠가 포함된 파일의 경로를 URL 제공합니다.
- 공백이나 특수 문자가 있는 파일을 사용하는 경우 터미널의 인용 및 이스케이프 규칙을 따릅니다.

> **Note**
>
> AWS CloudFormation 템플릿 를 식별하는 파라미터URL와 같이 이미 를 예상하는 파라미터의 경우 이 동작은 자동으로 비활성화됩니다URL. AWS CLI 구성 파일에서 `cli_follow_urlparam` 설정을 비활성화하여 이 동작을 비활성화할 수도 있습니다.

다음 예제의 파일 경로는 현재 작업 디렉터리를 기준으로 해석됩니다.

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"
```

`file://` 접두사 옵션은 "~/", "./", "../"를 포함한 Unix 스타일의 확장을 지원합니다. Windows에서는 "~/" 표현식이 %USERPROFILE% 환경 변수에 저장된 사용자 디렉터리로 확장합니다. 예를 들어, Windows 10은 일반적으로 `C:\Users\`*UserName*`\` 아래에 사용자 디렉터리가 있습니다.

다른 JSON 문서의 값으로 포함된 JSON 문서는 여전히 이스케이프해야 합니다.

```
$ 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\"}"
}
```

## 이진 파일

이진 데이터를 파라미터로 갖고 있는 명령의 경우 `fileb://` 접두사를 사용하여 데이터가 이진 콘텐츠임을 지정합니다. 이진 데이터를 수락하는 명령은 다음과 같습니다.

- **aws ec2 run-instances:** --user-data 파라미터.
- **aws s3api put-object:** --sse-customer-key 파라미터.
- **aws kms decrypt:** --ciphertext-blob 파라미터.

다음 예제에서는 Linux 명령줄 도구를 사용하여 바이너리 256비트 AES 키를 생성한 다음 Amazon S3에 제공하여 업로드된 파일 서버 측을 암호화합니다.

```
$ dd if=/dev/urandom bs=1 count=32 > sse.key
32+0 records in
32+0 records out
32 bytes (32 B) copied, 0.000164441 s, 195 kB/s
$ aws s3api put-object \
    --bucket amzn-s3-demo-bucket \
    --key test.txt \
    --body test.txt \
    --sse-customer-key fileb://sse.key \
    --sse-customer-algorithm AES256
{
    "SSECustomerKeyMD5": "iVg8oWa8sy714+FjtesrJg==",
    "SSECustomerAlgorithm": "AES256",
    "ETag": "\"a6118e84b76cf98bf04bbe14b6045c6c\""
}
```

### 원격 파일

AWS CLI 또한 는 http:// 또는 https:// 를 사용하여 인터넷에 호스팅된 파일에서 파라미터 로드를 지원합니다URL. 다음 예제에서는 Amazon S3 버킷에 저장된 파일을 참조합니다. 이렇게 하면 모든 컴퓨터에서 파라미터 파일에 액세스할 수 있지만, 컨테이너에 공개적으로 액세스할 수 있어야 합니다.

```
$ aws ec2 run-instances \
    --image-id ami-12345678 \
    --block-device-mappings http://amzn-s3-demo-bucket.s3.amazonaws.com/filename.json
```

이전 예제에서는 파일에 다음 JSON 데이터가 filename.json 포함되어 있다고 가정합니다.

```
[
  {
    "DeviceName": "/dev/sdb",
    "Ebs": {
      "VolumeSize": 20,
```

```
      "DeleteOnTermination": false,
      "VolumeType": "standard"
    }
  }
]
```

JSON형식이 지정된 파라미터가 포함된 파일을 참조하는 또 다른 예는 섹션을 참조하세요사용자에게 IAM 관리형 정책 연결.

## AWS CLI 의 스켈레톤 및 입력 파일 AWS CLI

대부분의 AWS CLI 명령은 파일에서 모든 파라미터 입력을 허용합니다. 이러한 템플릿은 `generate-cli-skeleton` 옵션을 사용하여 생성할 수 있습니다.

주제

- AWS CLI 스켈레톤 및 입력 파일 정보
- 명령 스켈레톤 생성

### AWS CLI 스켈레톤 및 입력 파일 정보

대부분의 AWS Command Line Interface (AWS CLI) 명령은 `--cli-input-json` 파라미터 를 사용하여 파일에서 모든 파라미터 입력을 수락하는 기능을 지원합니다.

이러한 동일한 명령은 편집하고 채울 수 있는 모든 `--generate-cli-skeleton` 파라미터가 포함된 JSON 형식의 파일을 생성하는 파라미터를 유용하게 제공합니다. 그러면 관련된 `--cli-input-json` 파라미터를 사용하여 명령을 실행하고 입력된 파일을 가리킬 수 있습니다.

> **Important**
>
> 여러 AWS CLI 명령은 명령과 같은 개별 AWS API 작업에 직접 매핑되지 않습니다`aws s3`. 이러한 명령은 이 주제에서 다루는 `--generate-cli-skeleton` 또는 `--cli-input-json` 파라미터를 지원하지 않습니다. 특정 명령이 이러한 파라미터를 지원하는지 여부를 모르는 경우 다음 명령을 실행하여 *service* 그리고 *command* 관심 있는 이름이 있는 이름입니다.
>
> ```
> $ aws service command help
> ```
>
> 지정된 명령이 지원하는 파라미터를 보여주는 Synopsis 섹션이 출력에 포함됩니다.
>
> ```
> $ aws iam list-users help
> ```

```
...
SYNOPSIS
          list-users
          ...
          [--cli-input-json]
          ...
          [--generate-cli-skeleton <value>]
...
```

`--generate-cli-skeleton` 파라미터를 사용하면 명령이 실행되지 않지만, 사용자 지정하여 이후 명령에 입력으로 사용할 수 있는 파라미터 템플릿을 생성하고 표시할 수 있습니다. 생성된 템플릿에는 명령이 지원하는 모든 파라미터가 포함됩니다.

`--generate-cli-skeleton` 파라미터는 다음 값 중 하나를 허용합니다.

- `input` - 생성된 템플릿에는 형식으로 지정된 모든 입력 파라미터가 포함됩니다JSON. 이것이 기본값입니다.
- `output` - 생성된 템플릿에는 형식으로 지정된 모든 출력 파라미터가 포함됩니다JSON.

AWS CLI 는 기본적으로 서비스의 에 대한 '래퍼'이므로 API스켈레톤 파일은 기본 파라미터 이름으로 모든 API 파라미터를 참조해야 합니다. 이는 AWS CLI 파라미터 이름과 다를 수 있습니다. 예를 들어 라는 AWS CLI 파라미터는 라는 AWS 서비스 API 파라미터에 매핑될 `user-name` 수 있습니다`UserName`(변경된 대문자 및 누락된 대시 표시). 실수를 방지하려면 `--generate-cli-skeleton` 옵션을 사용하여 “정확한” 파라미터 이름으로 템플릿을 생성하는 것이 좋습니다. 서비스의 API 참조 가이드를 참조하여 예상 파라미터 이름을 확인할 수도 있습니다. 템플릿에서 필요하지 않아 값을 지정하지 않을 파라미터를 모두 삭제할 수 있습니다.

예를 들어 다음 명령을 실행하면 Amazon Elastic Compute Cloud(Amazon EC2) 명령 에 대한 파라미터 템플릿이 생성됩니다run-instances.

JSON

다음 예제에서는 `--generate-cli-skeleton` 파라미터의 기본값(`input`)을 JSON 사용하여 형식이 지정된 템플릿을 생성하는 방법을 보여줍니다.

```
$ 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
}
```

## 명령 스켈레톤 생성

파라미터 스켈레톤 파일을 생성하고 사용하려면

1. `--generate-cli-skeleton` 파라미터로 명령을 실행하여 JSON 를 생성하고 출력을 파일로 전달하여 저장합니다.

   JSON

   ```
   $ aws ec2 run-instances --generate-cli-skeleton input > ec2runinst.json
   ```

2. 텍스트 편집기에서 파라미터 스켈레톤 파일을 열고 필요하지 않은 파라미터를 제거합니다. 예를 들어, 템플릿을 다음과 같이 줄일 수 있습니다. 파일이 여전히 유효한지 JSON 필요하지 않은 요소를 제거한 후에도 유효한지 확인하세요.

   JSON

   ```
   {
       "DryRun": true,
       "ImageId": "",
       "KeyName": "",
       "SecurityGroups": [
           ""
       ],
       "InstanceType": "",
       "Monitoring": {
           "Enabled": true
       }
   }
   ```

   이 예제에서는 Amazon EC2드라이 런 기능을 사용하도록 `DryRun` 파라미터를 `true`로 설정한 상태로 둡니다. 이 기능을 사용하면 실제로 리소스를 생성하거나 수정하지 않고도 명령을 안전하게 테스트할 수 있습니다.

3. 나머지 값을 시나리오에 적합한 값으로 채우십시오. 이 예제에서는 사용할 Amazon Machine Image(AMI)의 인스턴스 유형, 키 이름, 보안 그룹 및 식별자를 제공합니다. 이 예제에서는 기본 AWS 리전을 가정합니다. AMI `ami-dfc39aef` 는 `us-west-2` 리전에서 호스팅되는 64비트 Amazon Linux 이미지입니다. 다른 리전을 사용하는 경우 [를 사용하려면 올바른 AMI ID를 찾아](#)야 합니다.

   JSON

   ```
   {
       "DryRun": true,
       "ImageId": "ami-dfc39aef",
       "KeyName": "mykey",
       "SecurityGroups": [
           "my-sg"
       ],
       "InstanceType": "t2.micro",
       "Monitoring": {
           "Enabled": true
   ```

```
    }
}
```

4. file:// 접두사를 사용해 완료된 템플릿 파일을 --cli-input-json 파라미터로 전달하여 완료된 파라미터로 명령을 실행합니다. 는 경로를 현재 작업 디렉터리를 기준으로 AWS CLI 해석하므로 다음 예제에서는 경로가 없는 파일 이름만 표시하므로 현재 작업 디렉터리에서 파일을 직접 찾습니다.

   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.
   ```

   건식 실행 오류는 JSON 가 올바르게 형성되었고 파라미터 값이 유효함을 나타냅니다. 출력에 다른 문제가 보고되면 문제를 해결하고 "Request would have succeeded" 메시지가 표시될 때까지 이전 단계를 반복합니다.

5. 이제 테스트 실행을 비활성화하기 위해 DryRun 파라미터를 false로 설정할 수 있습니다.

   JSON

   ```
   {
       "DryRun": false,
       "ImageId": "ami-dfc39aef",
       "KeyName": "mykey",
       "SecurityGroups": [
           "my-sg"
       ],
       "InstanceType": "t2.micro",
       "Monitoring": {
           "Enabled": true
       }
   }
   ```

6. 명령을 실행하고 run-instances 실제로 Amazon EC2 인스턴스를 시작하고 성공적인 시작으로 생성된 세부 정보를 표시합니다. 출력 형식은 입력 파라미터 템플릿의 형식과 별도로 --output 파라미터에 의해 제어됩니다.

JSON

```
$ aws ec2 run-instances --cli-input-json file://ec2runinst.json --output json
```

```
{
    "OwnerId": "123456789012",
    "ReservationId": "r-d94a2b1",
    "Groups": [],
    "Instances": [
...
```

## 에서 간이 구문 사용 AWS CLI

AWS Command Line Interface (AWS CLI)는 JSON 형식의 많은 옵션 파라미터를 수락할 수 있습니다. 그러나 명령줄에 큰 JSON 목록이나 구조를 입력하는 것은 지루할 수 있습니다. 이를 더 쉽게 하기 위해 는 전체 JSON 형식을 사용하는 것보다 옵션 파라미터를 더 간단하게 표현할 수 있는 간략 구문 AWS CLI 도 지원합니다.

주제

- 구조 파라미터
- 에서 단축 구문 사용 AWS Command Line Interface

### 구조 파라미터

의 간이 구문을 AWS CLI 사용하면 사용자가 평면(중첩되지 않은 구조)인 파라미터를 더 쉽게 입력할 수 있습니다. 형식은 쉼표로 구분된 키 값 페어 목록입니다. 간편 구문은 문자열이므로 해당 터미널에 적용되는 인용 및 이스케이프 규칙을 사용해야 합니다.

Linux or macOS

```
--option key1=value1,key2=value2,key3=value3
```

PowerShell

```
--option "key1=value1,key2=value2,key3=value3"
```

둘 다 에서 형식이 지정된 다음 예제와 동일합니다JSON.

```
--option '{"key1":"value1","key2":"value2","key3":"value3"}'
```

쉼표로 구분된 각 키 값 페어 사이에 공백이 없어야 합니다. 다음은 update-table 옵션이 간편 방식으로 지정되어 있는 Amazon DynamoDB --provisioned-throughput 명령입니다.

```
$ aws dynamodb update-table \
    --provisioned-throughput ReadCapacityUnits=15,WriteCapacityUnits=10 \
    --table-name MyDDBTable
```

이는 에서 형식이 지정된 다음 예제와 동일합니다JSON.

```
$ aws dynamodb update-table \
    --provisioned-throughput '{"ReadCapacityUnits":15,"WriteCapacityUnits":10}' \
    --table-name MyDDBTable
```

## 에서 단축 구문 사용 AWS Command Line Interface

JSON 또는 간략이라는 두 가지 방법으로 목록 양식에서 입력 파라미터를 지정할 수 있습니다. AWS CLI 의 간편 구문은 숫자, 문자열 또는 비중첩 구조가 있는 목록을 더 쉽게 입력할 수 있도록 하기 위해 설계되었습니다.

기본 형식은 여기에 표시됩니다. 여기서 목록의 값은 단일 공백으로 구분됩니다.

```
--option value1 value2 value3
```

이는 에서 형식이 지정된 다음 예제와 동일합니다JSON.

```
--option '[value1,value2,value3]'
```

앞에서 언급한 바와 같이, 숫자 목록, 문자열 목록 또는 비중첩 구조 목록을 간편 방식으로 지정할 수 있습니다. 다음은 Amazon Elastic Compute Cloud(AmazonEC2)에 대한 stop-instances 명령의 예입니다. 여기서 --instance-ids 옵션에 대한 입력 파라미터(스트링 목록)는 요약으로 지정됩니다.

```
$ aws ec2 stop-instances \
    --instance-ids i-1486157a i-1286157c i-ec3a7e87
```

이는 에서 형식이 지정된 다음 예제와 동일합니다JSON.

```
$ aws ec2 stop-instances \
    --instance-ids '["i-1486157a","i-1286157c","i-ec3a7e87"]'
```

다음 예제에서는 `--tags` 옵션에 중첩되지 않은 구조 목록을 가져오는 Amazon EC2 `create-tags` 명령을 보여줍니다. `--resources` 옵션은 태깅할 인스턴스의 ID를 지정합니다.

```
$ aws ec2 create-tags \
    --resources i-1286157c \
    --tags Key=My1stTag,Value=Value1 Key=My2ndTag,Value=Value2
 Key=My3rdTag,Value=Value3
```

이는 에서 형식이 지정된 다음 예제와 동일합니다JSON. JSON 파라미터는 가독성을 위해 여러 줄에 기록됩니다.

```
$ aws ec2 create-tags \
    --resources i-1286157c \
    --tags '[
        {"Key": "My1stTag", "Value": "Value1"},
        {"Key": "My2ndTag", "Value": "Value2"},
        {"Key": "My3rdTag", "Value": "Value3"}
    ]'
```

# 에서 명령 출력 제어 AWS CLI

이 단원에서는 AWS Command Line Interface (AWS CLI)의 출력을 제어하는 다양한 방법에 대해 설명합니다. 터미널에서 AWS CLI 출력을 사용자 지정하면 가독성을 높이고, 스크립팅 자동화를 간소화하고, 더 큰 데이터 세트를 통해 더 쉽게 탐색할 수 있습니다.

는 , , 및 를 포함한 여러 출력 형식을 AWS CLI 지원합니다`text`. `table` `json` 일부 서비스에는 데이터에 대한 서버 측 페이지 매김이.

마지막으로 AWS CLI 에는 출력을 개별적으로 또는 함께 필터링하는 데 사용할 수 있는 서버 측 및 클라이언트 측 필터링이 모두 있습니다 AWS CLI .

주제

- 민감한 출력
- 서버 측 출력 옵션과 클라이언트 측 출력 옵션 비교
- 에서 출력 형식 설정 AWS CLI

- 의 페이지 매김 옵션 사용 AWS CLI
- 에서 출력 필터링 AWS CLI

## 민감한 출력

의 일부 작업은 환경 변수의 정보를 포함하여 민감한 것으로 간주될 수 있는 정보를 반환할 수 AWS CLI 있습니다. 이 정보의 노출은 특정 시나리오에서 보안 위험을 나타낼 수 있습니다. 예를 들어, 이 정보는 지속적 통합 및 지속적 배포(CI/CD) 로그에 포함될 수 있습니다. 따라서 이러한 출력을 로그의 일부로 포함할 때 검토하고 필요하지 않을 때는 출력을 억제하는 것이 중요합니다.

민감한 데이터 보호에 대한 자세한 내용은 섹션을 참조하세요the section called “데이터 보호”.

다음 모범 사례를 고려하세요.

- 와 같은 보안 암호 스토어에서 암호를 프로그래밍 방식으로 검색하는 것이 좋습니다 AWS Secrets Manager.
- 빌드 로그의 내용을 검토하여 민감한 정보가 포함되어 있지 않은지 확인합니다. 출력에 대한 파이핑 `/dev/null` 또는 출력을 bash 또는 PowerShell 변수로 캡처하는 등의 접근 방식을 고려하여 명령 출력을 억제합니다.

  다음은 오류가 아닌 출력을 로 리디렉션하기 위한 bash 예제입니다`/dev/null`.

  ```
  $ aws s3 ls > /dev/null
  ```

  터미널의 출력 억제에 대한 자세한 내용은 사용하는 터미널의 사용 설명서를 참조하세요.
- 로그의 액세스를 고려하고 사용 사례에 맞게 액세스 범위를 적절히 지정합니다.

## 서버 측 출력 옵션과 클라이언트 측 출력 옵션 비교

AWS CLI 에는 개별적으로 또는 함께 사용하여 출력을 필터링할 수 있는 서버 측 및 클라이언트 측 필터링이 모두 있습니다 AWS CLI . 서버 측 필터링이 먼저 처리되고 클라이언트 측 필터링에 대한 출력을 반환합니다. 서버 측 필터링은 서비스 에서 지원됩니다API. 클라이언트 측 필터링은 `--query` 파라미터를 사용하여 AWS CLI 클라이언트에서 지원됩니다.

서버 측 출력 옵션은 에서 직접 지원하는 기능입니다 AWS 서비스 API. 필터링되거나 호출된 모든 데이터는 클라이언트로 전송되지 않으므로 HTTP 응답 시간을 단축하고 더 큰 데이터 세트의 대역폭을 개선할 수 있습니다.

클라이언트 측 출력 옵션은 AWS CLI에서 만든 기능입니다. 모든 데이터가 클라이언트로 전송된 다음 표시된 콘텐츠를 AWS CLI 필터링하거나 페이지합니다. 클라이언트 측 작업으로는 대규모 데이터 세트의 속도나 대역폭이 절약되지 않습니다.

서버 측 옵션과 클라이언트 측 옵션을 함께 사용하면 서버 측 작업이 먼저 완료된 후 클라이언트로 전송되어 클라이언트 측 작업이 진행됩니다. 이를 통해 서버 측 옵션의 잠재적인 속도 및 대역폭 절감 효과를 활용하는 동시에 추가 AWS CLI 기능을 사용하여 원하는 결과를 얻을 수 있습니다.

## 에서 출력 형식 설정 AWS CLI

이 주제에서는 AWS Command Line Interface ()의 다양한 출력 형식을 설명합니다AWS CLI. AWS CLI은(는) 다음 출력 형식을 지원합니다.

- **`json`** - 출력은 JSON 문자열 형식입니다.
- **`text`** - 출력은 여러 줄의 탭으로 구분된 문자열 값으로 형식이 지정됩니다. 출력을 `grep`, `sed` 또는 `awk`와 같은 텍스트 프로세서로 전달하는 데 사용할 수 있습니다.
- **`table`** - 출력은 셀 테두리를 형성하기 위해 +|- 문자를 사용하여 표로 형식이 지정됩니다. 일반적으로 읽기는 쉽지만 프로그래밍 방식으로는 유용하지 않은 "인간 친화적" 형식으로 정보를 표시합니다.

### 출력 형식을 선택하는 방법

구성 주제의 설명과 같이, 다음 세 가지 방법으로 출력 형식을 지정할 수 있습니다.

- **`output`** 파일의 명명된 프로파일에서 **`config`** 옵션 사용 - 다음 예제에서는 기본 출력 형식을 `text`로 설정합니다.

```
[default]
output=text
```

- **`AWS_DEFAULT_OUTPUT`** 환경 변수 사용 - 다음 출력은 변수가 변경되거나 세션이 끝날 때까지 이 명령줄 세션의 명령 형식을 `table`로 지정합니다. 이 환경 변수를 사용하면 `config` 파일에 설정된 값을 재정의합니다.

```
$ export AWS_DEFAULT_OUTPUT="table"
```

- 명령줄에서 **`--output`** 옵션 사용 - 다음 예제에서는 이 명령의 출력만 `json`으로 설정합니다. 명령에 이 옵션을 사용하면 현재 설정된 환경 변수 또는 `config` 파일의 값을 재정의합니다.

```
$ aws swf list-domains --registration-status REGISTERED --output json
```

**Important**

지정한 출력 유형에 따라 `--query` 옵션 작동 방식이 변경됩니다.

- `--output text`를 지정하면 `--query` 필터가 적용되기 전에 출력이 페이지 매김되고 는 출력의 각 페이지에서 쿼리를 한 번 AWS CLI 실행합니다. 이로 인해, 쿼리에는 예상치 못한 추가 출력이 발생할 수 있는 각 페이지의 첫 번째 일치하는 요소가 포함됩니다. 출력을 추가로 필터링하려면 `head` 또는 `tail` 등 다른 명령줄 도구를 사용할 수 있습니다.
- `--output json`,을 지정하면 해당 출력을 하나의 네이티브 구조로 완전히 처리한 뒤에 `--query` 필터를 적용합니다. 는 전체 구조에 대해 쿼리를 한 번만 AWS CLI 실행하여 필터링된 결과를 생성한 다음 출력합니다.

## JSON 출력 형식

JSON 는 의 기본 출력 형식입니다 AWS CLI. 대부분의 프로그래밍 언어는 내장 함수 또는 공개적으로 사용 가능한 라이브러리를 사용하여 JSON 문자열을 쉽게 디코딩할 수 있습니다. 형식이 지정된 JSON 출력을 필터링하고 형식을 지정하는 강력한 방법으로 출력을 - AWS CLI JSON-쿼리 옵션과 결합할 수 있습니다.

로 수행할 수 없는 고급 필터링의 경우 명령줄 JSON 프로세서`jq`인 를 고려할 `--query`수 있습니다. http://stedolan.github.io/jq/에서 이 처리기를 다운로드하고 공식 자습서를 찾아볼 수 있습니다.

다음은 JSON 출력의 예입니다.

```
$ aws iam list-users --output json
```

```
{
    "Users": [
        {
            "Path": "/",
            "UserName": "Admin",
            "UserId": "AIDA1111111111EXAMPLE",
            "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": "AIDA2222222222EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:user/backup/backup-user",
            "CreateDate": "2019-09-17T19:30:40+00:00"
        },
        {
            "Path": "/",
            "UserName": "cli-user",
            "UserId": "AIDA3333333333EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:user/cli-user",
            "CreateDate": "2019-09-17T19:11:39+00:00"
        }
    ]
}
```

## 텍스트 출력 형식

`text` 형식은 AWS CLI 출력을 탭으로 구분된 줄로 구성합니다. `grep`, 및 `sed`와 같은 기존 Unix 텍스트 도구 `awk`및 에서 수행하는 텍스트 처리와 함께 사용할 수 있습니다 PowerShell.

`text` 출력 형식은 아래와 같은 기본 구조를 따릅니다. 열은 기본 JSON 객체의 해당 키 이름을 기준으로 알파벳순으로 정렬됩니다.

```
IDENTIFIER  sorted-column1 sorted-column2
IDENTIFIER2 sorted-column1 sorted-column2
```

다음은 `text` 출력의 예제입니다. 각 필드는 다른 항목과 구분된 탭이며 빈 필드가 있는 추가 탭이 있습니다.

```
$ 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   /          AIDA1111111111EXAMPLE   Admin
USERS   arn:aws:iam::123456789012:user/backup/backup-user   2019-09-17T19:30:40+00:00
                            /backup/   AIDA2222222222EXAMPLE   backup-user
USERS   arn:aws:iam::123456789012:user/cli-user             2019-09-17T19:11:39+00:00
                            /          AIDA3333333333EXAMPLE   cli-user
```

네 번째 열은 PasswordLastUsed 필드이며, 해당 사용자가 AWS Management Console콘솔에 로그인하지 않기 때문에 마지막 두 항목은 비어 있습니다.

> **Important**
>
> *text* 출력을 지정하는 경우 일관된 동작을 보장하기 위해 항상 *--query* 옵션도 사용하는 것이 좋습니다.
> 이는 텍스트 형식이 AWS 서비스에서 반환한 기본 JSON 객체의 키 이름으로 출력 열을 알파벳순으로 정렬하고 유사한 리소스의 키 이름이 동일하지 않을 수 있기 때문입니다. 예를 들어 Linux 기반 Amazon EC2 인스턴스의 JSON 표현에는 Windows 기반 인스턴스의 JSON 표현에 존재하지 않는 요소가 있거나 그 반대일 수 있습니다. 또한 리소스에는 향후 업데이트에서 추가되거나 제거되어 열 순서를 변경하는 키-값 요소가 있을 수 있습니다. 이러한 경우 --query를 사용하면 출력 형식을 완전히 제어할 수 있도록 text 출력의 기능이 향상됩니다.
> 다음 예제에서 명령은 표시할 요소를 지정하고 목록 표기법 [key1, key2, ...]를 사용하여 열의 순서를 정의합니다. 이렇게 하면 예상 열에 올바른 키 값이 항상 표시된다는 완전한 확신을 사용자에게 제공할 수 있습니다. 마지막으로 AWS CLI 출력이 존재하지 않는 키None의 값으로 어떻게 표시되는지 알아봅니다.
>
> ```
> $ 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   AIDA1111111111EXAMPLE
> backup-user   arn:aws:iam::123456789012:user/backup-user
>  2019-09-17T19:30:40+00:00   None                        AIDA2222222222EXAMPLE
> cli-user      arn:aws:iam::123456789012:user/cli-backup
>  2019-09-17T19:11:39+00:00   None                        AIDA3333333333EXAMPLE
> ```

다음 예제는 grep 명령의 awk 출력에 text 및 aws ec2 describe-instances를 사용하는 방법을 보여줍니다. 첫 번째 명령은 text 출력에 각 인스턴스의 가용 영역, 현재 상태 및 인스턴스 ID를 표시합니다. 두 번째 명령은 를 출력하는 프로세스를 처리하여 us-west-2a 가용 영역에서 실행 중인 모든 인스턴스IDs의 인스턴스만 표시합니다.

```
$ 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
```

다음 예제는 한 단계 더 나아가 출력을 필터링하는 방법뿐만 아니라 출력을 사용하여 중지된 각 인스턴스의 인스턴스 유형 변경을 자동화하는 방법을 보여줍니다.

```
$ 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
```

`text` 출력은 에서도 유용할 수 있습니다 PowerShell. `text` 출력의 열은 탭으로 구분되므로 PowerShell의 `` `t `` 구분 기호를 사용하여 출력을 배열로 쉽게 분할할 수 있습니다. 다음 명령은 첫 번째 열(`InstanceId`)이 `AvailabilityZone` 문자열과 일치할 경우 세 번째 열(`us-west-2a`)의 값을 표시합니다.

```
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
```

이전 예제에서는 `--query` 파라미터를 사용하여 기본 JSON 객체를 구문 분석하고 원하는 열을 가져오는 방법을 보여주지만, 교차 플랫폼 호환성이 문제가 되지 않는 JSON경우 를 처리할 수 있는 자체 기능이 PowerShell 있습니다. 대부분의 명령 쉘에서 요구하는 대로 출력을 텍스트로 처리하는 대신 `ConvertFrom-JSON` cmdlet을 사용하여 계층 구조화된 객체를 생성할 수 PowerShell 있습니다. 그런 다음 해당 객체에서 직접 원하는 멤버에 액세스할 수 있습니다.

```
(aws ec2 describe-instances --output json | ConvertFrom-
Json).Reservations.Instances.InstanceId
```

**Tip**

텍스트를 출력하고 `--query` 파라미터를 사용하여 단일 필드로 필터링하는 경우, 탭으로 구분된 값이 한 줄로 출력됩니다. 각 값을 별개의 줄로 가져오려면 다음 예제에 표시된 대로 출력 필드를 괄호 안에 넣으면 됩니다.
탭으로 구분되어 한 줄로 출력:

```
$ aws iam list-groups-for-user --user-name susan  --output text --query
 "Groups[].GroupName"
```

```
HRDepartment    Developers      SpreadsheetUsers  LocalAdmins
```

`[GroupName]`을 괄호 안에 넣어서 각 값을 자체의 줄에 출력:

```
$ aws iam list-groups-for-user --user-name susan  --output text --query
 "Groups[].[GroupName]"
```

```
HRDepartment
Developers
SpreadsheetUsers
LocalAdmins
```

## 테이블 출력 형식

`table` 형식은 사람이 읽을 수 있는 복잡한 AWS CLI 출력 표시를 표 형식으로 생성합니다.

```
$ 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 | /        | AIDA1111111111EXAMPLE | Admin       ||
||  arn:aws:iam::123456789012:user/backup/backup-user | 2019-09-17T19:30:40+00:00 |
                        | /backup/ | AIDA2222222222EXAMPLE | backup-user ||
||  arn:aws:iam::123456789012:user/cli-user           | 2019-09-17T19:11:39+00:00 |
                        | /        | AIDA3333333333EXAMPLE | cli-user    ||
+-----------------------------------------------------------------------------------------------------
+
```

--query 옵션을 table 형식과 결합하여 원시 출력에서 미리 선택한 요소 집합을 표시할 수 있습니다. 사전 표기법과 목록 표기법의 출력 차이에 주의하십시오. 첫 번째 예제에서는 열 이름이 알파벳 순서로 정렬되고 두 번째 예제에서는 이름 없는 열이 사용자가 정의한 방식으로 정렬됩니다. --query 옵션에 대한 자세한 내용은 에서 출력 필터링 AWS CLI 단원을 참조하십시오.

```
$ 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  |
+--------------+--------------+--------------+-----+
```

# 의 페이지 매김 옵션 사용 AWS CLI

이 주제에서는 AWS CLI의 출력에 페이지 번호를 매기는 다양한 방법에 대해 설명합니다.

## 서버 측 페이지 매김

많은 항목 목록을 반환할 수 있는 명령의 경우 AWS Command Line Interface (AWS CLI)에는 가 서비스를 AWS CLI 호출API하여 목록을 채울 때 출력에 포함된 항목 수를 제어하는 여러 옵션이 있습니다.

옵션에는 다음 사항이 포함됩니다.

- --no-paginate 파라미터를 사용하는 방법
- --page-size 파라미터를 사용하는 방법
- --max-items 파라미터를 사용하는 방법
- --starting-token 파라미터를 사용하는 방법

기본적으로 는 개별 서비스에 의해 결정된 페이지 크기를 AWS CLI 사용하고 사용 가능한 모든 항목을 검색합니다. 예를 들어 Amazon S3의 기본 페이지 크기는 1,000입니다. 3,500개 객체를 포함하는 Amazon S3 버킷에서 `aws s3api list-objects`를 실행할 경우 AWS CLI 는 백그라운드에서 서비스별 페이지 매김 로직을 처리하고 최종 출력에 3,500개 객체를 모두 반환하면서 Amazon S3에 대한 4개 호출을 자동으로 작성합니다.

--no-paginate 파라미터를 사용하는 방법

`--no-paginate` 옵션은 클라이언트 측에서 다음 페이지 매김 토큰을 사용 중지합니다. 명령을 사용할 때 기본적으로 는 AWS CLI 자동으로 여러 번 호출하여 페이지 매김을 생성하는 데 가능한 모든 결과를 반환합니다. 각 페이지에 대해 한 번 호출합니다. 페이지 매김을 비활성화하면 명령 결과의 첫 페이지에 대해 AWS CLI 한 번만 호출됩니다.

예를 들어 객체가 3,500개 포함된 Amazon S3 버킷`aws s3api list-objects`에서 를 실행하는 경우 는 Amazon S3에 대한 첫 번째 호출 AWS CLI 만 수행하여 최종 출력에서 처음 1,000개 객체만 반환합니다.

```
$ aws s3api list-objects \
    --bucket amzn-s3-demo-bucket \
    --no-paginate
{
    "Contents": [
...
```

--page-size 파라미터를 사용하는 방법

많은 리소스에서 list 명령을 실행할 때 문제가 발생할 경우, 기본값 페이지 크기가 너무 크기 때문일 수 있습니다. 이로 인해 AWS 서비스에 대한 호출이 최대 허용 시간을 초과하고 "시간 초과" 오류가 발생할 수 있습니다. `--page-size` 옵션을 사용하여 AWS 서비스에 대한 각 호출에서 더 적은 수의 항목을 AWS CLI 요청하도록 지정할 수 있습니다. 는 AWS CLI 여전히 전체 목록을 검색하지만 백그라운드에서 더 많은 수의 서비스 API 호출을 수행하고 각 호출에서 더 적은 수의 항목을 검색합니다. 그러면 각각의 호출이 시간 초과 없이 성공할 확률이 높아집니다. 페이지 크기를 변경해도 출력에는 영향을 주지 않으며, 출력을 생성하기 위해 수행해야 하는 API 호출 수에만 영향을 미칩니다.

```
$ aws s3api list-objects \
    --bucket amzn-s3-demo-bucket \
    --page-size 100
{
    "Contents": [
...
```

--max-items 파라미터를 사용하는 방법

AWS CLI 출력에 한 번에 더 적은 항목을 포함하려면 `--max-items` 옵션을 사용합니다. 는 AWS CLI 여전히 앞서 설명한 대로 서비스의 페이지 매김을 처리하지만 지정한 시간에 항목 수만 출력합니다.

```
$ aws s3api list-objects \
    --bucket amzn-s3-demo-bucket \
    --max-items 100
{
    "NextToken": "eyJNYXJrZXIiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxfQ==",
    "Contents": [
...
```

--starting-token 파라미터를 사용하는 방법

출력 항목 수(--max-items)가 기본 API 호출에서 반환한 총 항목 수보다 적으면 출력에는 다음 항목 세트를 검색하기 위해 후속 명령에 전달할 수 NextToken 있는 이 포함됩니다. 다음 예제를 통해 앞의 예제에서 반환된 NextToken 값을 사용하는 방법을 배우고, 두 번째 백 개 항목을 검색할 수 있습니다.

> **Note**
>
> --starting-token 파라미터는 null이거나 비어있을 수 없습니다. 이전 명령이 NextToken 값을 반환하지 않으면 반환할 더 이상의 항목이 없는 것이기 때문에 명령을 다시 호출할 필요가 없습니다.

```
$ aws s3api list-objects \
    --bucket amzn-s3-demo-bucket \
    --max-items 100 \
    --starting-token eyJNYXJrZXIiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxfQ==
{
    "Contents": [
...
```

지정된 AWS 서비스는 호출할 때마다 동일한 순서로 항목을 반환하지 않을 수 있습니다. --page-size 및 --max-items에 서로 다른 값을 지정하면 누락되거나 중복된 항목을 포함해 예상치 못한 결과가 발생할 수 있습니다. 이를 방지하려면 --page-size 및 --max-items에 동일한 번호를 사용하여 AWS CLI 의 페이지 매김을 기본 서비스의 페이지 매김과 동기화하십시오. 또한 전체 목록을 검색하고 필요한 구문 분석 작업을 로컬에서 수행할 수 있습니다.

## 에서 출력 필터링 AWS CLI

AWS Command Line Interface (AWS CLI)에는 개별적으로 또는 함께 사용하여 AWS CLI 출력을 필터링할 수 있는 서버 측 및 클라이언트 측 필터링이 모두 있습니다. 서버 측 필터링이 먼저 처리되고 클라이언트 측 필터링에 대한 출력을 반환합니다.

- 서버 측 필터링은 에서 지원API되며 일반적으로 --filter 파라미터로 구현합니다. 서비스는 대규모 데이터 세트의 HTTP 응답 시간을 단축할 수 있는 일치하는 결과만 반환합니다.
- 클라이언트 측 필터링은 --query 파라미터를 사용하여 AWS CLI 클라이언트에서 지원됩니다. 이 파라미터에는 서버 측 필터링에 없을 수 있는 기능이 있습니다.

주제

## 서버 측 필터링

의 서버 측 필터링 AWS CLI 은 AWS 서비스 에서 제공합니다API. 서비스는 AWS 필터와 일치하는 HTTP 응답의 레코드만 반환하므로 대용량 데이터 세트의 HTTP 응답 시간을 단축할 수 있습니다. 서버 측 필터링은 서비스 에 의해 정의되므로 파라미터 이름과 함수API는 서비스마다 다릅니다. 필터링에 사용되는 몇 가지 일반적인 매개 변수 이름은 다음과 같습니다.

- `--filter`(예: ses 및 ce).
- `--filters`(예: ec2, autoscaling 및 rds).
- `filter`라는 단어로 시작하는 이름입니다(예: `aws dynamodb scan` 명령의 경우 `--filter-expression`).

특정 명령에 서버 측 필터링 및 필터링 규칙이 있는지에 대한 자세한 내용은 AWS CLI 참조 가이드 를 참조하세요.

## 클라이언트 측 필터링

는 `--query` 파라미터와 함께 기본 제공 JSON기반 클라이언트 측 필터링 기능을 AWS CLI 제공합니다. `--query` 매개 변수는 출력의 내용과 스타일을 사용자 지정하는 데 사용할 수 있는 강력한 도구입니다. `--query` 파라미터는 서버에서 반환되는 HTTP 응답을 가져와 결과를 표시하기 전에 필터링합니다. 전체 HTTP 응답은 필터링 전에 클라이언트로 전송되므로 클라이언트 측 필터링은 대규모 데이터 세트에 대한 서버 측 필터링보다 느릴 수 있습니다.

쿼리는 JMESPath 구문을 사용하여 출력을 필터링하기 위한 표현식을 생성합니다. 구문을 알아보려면 JMESPath 웹 사이트의 JMESPath 자습서를 참조하세요.

> **Important**
>
> 지정한 출력 유형에 따라 `--query` 옵션 작동 방식이 변경됩니다.
>
> - `--output text`를 지정하면 `--query` 필터가 적용되기 전에 출력이 페이지 매김되고 는 출력의 각 페이지에서 쿼리를 한 번 AWS CLI 실행합니다. 이로 인해, 쿼리에는 예상치 못한

추가 출력이 발생할 수 있는 각 페이지의 첫 번째 일치하는 요소가 포함됩니다. 출력을 추가로 필터링하려면 `head` 또는 `tail` 등 다른 명령줄 도구를 사용할 수 있습니다.

- `--output json`,을 지정하면 해당 출력을 하나의 네이티브 구조로 완전히 처리한 뒤에 `--query` 필터를 적용합니다. 는 전체 구조에 대해 쿼리를 한 번만 AWS CLI 실행하여 필터링된 결과를 생성한 다음 출력합니다.

클라이언트 측 필터링 주제



## 시작하기 전에

이 예제에서 사용된 필터 표현식을 사용할 때는 터미널 셸에 올바른 인용 규칙을 사용해야 합니다. 자세한 내용은 the section called “문자열과 따옴표” 단원을 참조하십시오.

다음 JSON 출력은 `--query` 파라미터가 생성할 수 있는 항목의 예를 보여줍니다. 출력은 별도의 Amazon EC2 인스턴스에 연결된 세 개의 Amazon EBS 볼륨을 설명합니다.

출력 예시

```
$ 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
    }
  ]
}
```

식별자

식별자는 출력 값의 레이블입니다. 필터를 만들 때 식별자를 사용하여 쿼리 결과 범위를 좁힙니다. 다음 출력 예제에서는 `Volumes`, `AvailabilityZone`, `AttachTime` 등 모든 식별자가 강조 표시됩니다.

```
$ 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
    }
  ]
}
```


자세한 내용은 JMESPath 웹 사이트의 식별자를 참조하세요.

목록에서 선택

목록 또는 배열은 [의 Volumes 및 Attachments 등 대괄호 “the section called “시작하기 전에”” 뒤에 오는 식별자입니다.

구문

```
<listName>[ ]
```

배열의 모든 출력을 필터링하려면 와일드 카드 표기법을 사용할 수 있습니다. 와일드카드 표현식은 * 표기법을 사용하여 요소를 반환하는 데 사용되는 표현식입니다.

다음 예제에서는 모든 Volumes 내용을 쿼리합니다.

```
$ 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
  }
]
```

인덱스별로 배열의 특정 볼륨을 보려면 배열 인덱스를 호출합니다. 예를 들어, `Volumes` 배열의 첫 번째 항목은 인덱스가 `0`이고 `Volumes[0]` 쿼리가 생성됩니다. 배열 인덱스에 대한 자세한 내용은 JMESPath 웹 사이트의 인덱스 표현식을 참조하세요.

```
$ 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
}
```

인덱스별로 특정 범위의 볼륨을 보려면 다음 구문과 함께 `slice`를 사용합니다. 여기서 start는 시작 배열 인덱스이고, stop은 필터가 처리를 중지하는 인덱스이고, step은 건너뛰기 간격입니다.

구문

```
<arrayName>[<start>:<stop>:<step>]
```

다음 항목 중 하나라도 슬라이스 표현식에서 생략된 경우, 다음 기본값을 사용합니다.

- 시작 - 목록의 첫 번째 인덱스, 0.
- 중지 - 목록의 마지막 인덱스.
- 단계 - 건너뛰기 단계 없음. 여기서 값은 1입니다.

처음 두 볼륨만 반환하려면 다음 예제와 같이 시작 값 0, 중지 값 2, 단계 값 1을 사용합니다.

```
$ 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
  }
]
```

이 예제에는 기본값이 포함되어 있으므로 슬라이스를 `Volumes[0:2:1]`에서 `Volumes[:2]`(으)로 줄일 수 있습니다.

다음 예제에서는 기본값을 생략하고 전체 배열에서 두 볼륨마다 반환합니다.

```
$ 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
  }
]
```

단계는 다음 예제와 같이 배열의 역순으로 필터링하는 음수를 사용할 수도 있습니다.

```
$ 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
  }
]
```

자세한 내용은 JMESPath 웹 사이트의 슬라이스를 참조하세요.

중첩된 데이터 필터링

중첩된 값의 `Volumes[*]` 필터링 범위를 좁히려면 마침표 및 필터 기준을 추가하여 하위 표현식을 사용합니다.

구문

```
<expression>.<expression>
```

다음 예제에서는 모든 볼륨에 대한 모든 `Attachments` 정보를 보여줍니다.

```
$ 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"
    }
  ]
]
```

중첩된 값으로 추가로 필터링하려면 중첩된 각 식별자에 대한 표현식을 추가합니다. 다음 예제에서는 모든 `State`에 대한 `Volumes`이(가) 나와 있습니다.

```
$ aws ec2 describe-volumes \
    --query 'Volumes[*].Attachments[*].State'
[
  [
    "attached"
  ],
  [
    "attached"
  ],
  [
    "attached"
  ]
]
```

결과 병합

자세한 내용은 JMESPath 웹 사이트의 섹션을 참조SubExpressions하세요.

와일드카드 표기법을 제거함으로써 `Volumes[*].Attachments[*].State`에 대한 결과를 병합하여 `Volumes[*].Attachments[].State` 쿼리를 생성할 수 있습니다. 병합은 흔히 결과의 가독성을 높이는 데 유용합니다.

```
$ aws ec2 describe-volumes \
    --query 'Volumes[*].Attachments[].State'
[
  "attached",
  "attached",
  "attached"
]
```

자세한 내용은 JMESPath 웹 사이트의 플랫팅을 참조하세요.

특정 값에 대한 필터링

목록의 특정 값을 필터링하려면 다음 구문과 같이 필터 표현식을 사용합니다.

구문

```
? <expression> <comparator> <expression>]
```

표현식 비교기에는 ==, !=, <, <=, >, >=이(가) 포함됩니다. 다음 예제에서는 `VolumeIdsVolumes`의 모든 `Attached`에 대해 `State`을(를) 필터링합니다.

```
$ aws ec2 describe-volumes \
    --query 'Volumes[*].Attachments[?State==`attached`].VolumeId'
[
  [
    "vol-e11a5288"
  ],
  [
    "vol-2e410a47"
  ],
  [
    "vol-a1b3c7nd"
  ]
]
```

그런 다음 이를 병합하여 다음 예제처럼 되게 할 수 있습니다.

```
$ aws ec2 describe-volumes \
    --query 'Volumes[*].Attachments[?State==`attached`].VolumeId[]'
[
  "vol-e11a5288",
  "vol-2e410a47",
  "vol-a1b3c7nd"
]
```

다음 예제에서는 크기가 20보다 작은 모든 `VolumeIds`의 `Volumes`을(를) 필터링합니다.

```
$ aws ec2 describe-volumes \
    --query 'Volumes[?Size < `20`].VolumeId'
[
  "vol-2e410a47",
  "vol-a1b3c7nd"
]
```

자세한 내용은 JMESPath 웹 사이트의 표현식 필터링을 참조하세요.

파이핑 표현식

필터 결과를 새 목록으로 파이핑한 후, 다음 구문을 사용하여 다른 표현식으로 결과를 필터링할 수 있습니다.

구문

```
<expression> | <expression>]
```

다음 예제에서는 `Volumes[*].Attachments[].InstanceId` 표현식의 필터 결과를 가져와 배열의 첫 번째 결과를 출력합니다.

```
$ aws ec2 describe-volumes \
    --query 'Volumes[*].Attachments[].InstanceId | [0]'
"i-a071c394"
```

이 예제는 먼저 다음 표현식에서 배열을 생작성하여 이 작업을 수행합니다.

```
$ aws ec2 describe-volumes \
    --query 'Volumes[*].Attachments[].InstanceId'
```

```
"i-a071c394",
  "i-4b41a37c",
  "i-1jd73kv8"
```

그런 다음 해당 배열의 첫 번째 요소를 반환합니다.

```
"i-a071c394"
```

자세한 내용은 JMESPath 웹 사이트의 파이프 표현식을 참조하세요.

여러 식별자 값에 대한 필터링

여러 식별자를 필터링하려면 다음 구문을 사용하여 다중 선택 목록을 사용합니다.

구문

```
<listName>[].[<expression>, <expression>]
```

다음 예제에서는 `VolumeId` 및 `VolumeType`이(가) `Volumes` 목록에서 필터링되어 다음 표현식이 생성됩니다.

```
$ aws ec2 describe-volumes \
    --query 'Volumes[].[VolumeId, VolumeType]'
[
  [
    "vol-e11a5288",
    "standard"
  ],
  [
    "vol-2e410a47",
    "standard"
  ],
  [
    "vol-a1b3c7nd",
    "standard"
  ]
]
```

중첩된 데이터를 목록에 추가하려면 다른 다중 선택 목록을 추가합니다. 다음 예제에서는 중첩 `InstanceId` 목록에서 `State` 및 `Attachments`도 필터링하여 이전 예제를 확장합니다. 그러면 다음과 같은 표현식이 생성됩니다.

```
$ 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"
      ]
    ]
  ]
]
```

더 읽기 쉽게 하려면 다음 예제와 같이 표현식을 병합합니다.

```
$ 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"
  ]
]
```

자세한 내용은 JMESPath 웹 사이트의 다중 선택 목록을 참조하세요.

식별자 값에 레이블 추가

이 출력을 더 읽기 쉽게 하려면 다음 구문을 사용하여 다중 선택 해시를 사용합니다.

구문

```
<listName>[].{<label>: <expression>, <label>: <expression>}
```

식별자 레이블이 식별자 이름과 같을 필요는 없습니다. 다음 예제에서는 `VolumeType` 값에 `VolumeType` 레이블을 사용합니다.

```
$ aws ec2 describe-volumes \
    --query 'Volumes[].{VolumeType: VolumeType}'
[
  {
    "VolumeType": "standard",
  },
  {
    "VolumeType": "standard",
  },
  {
    "VolumeType": "standard",
  }
```

```
]
```

간단히, 다음 예제에서는 각 레이블의 식별자 이름을 유지하고 모든 볼륨에 대해 `VolumeId`, `VolumeType`, `InstanceId` 및 `State`을(를) 표시합니다.

```
$ 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"
  }
]
```

자세한 내용은 JMESPath 웹 사이트의 다중 선택 해시를 참조하세요.

함수

JMESPath 구문에는 쿼리에 사용할 수 있는 많은 함수가 포함되어 있습니다. JMESPath 함수에 대한 자세한 내용은 JMESPath 웹 사이트의 기본 제공 함수를 참조하세요.

함수를 쿼리에 통합하는 방법을 보여주기 위해 다음 예제에서는 `sort_by` 함수를 사용합니다. `sort_by` 함수는 다음 구문을 사용하여 표현식을 정렬 키로 사용해 배열을 정렬합니다.

구문

```
sort_by(<listName>, <sort expression>)[].<expression>
```

다음 예제에서는 이전의 다중 선택 해시 예제를 사용하고 VolumeId(으)로 출력을 정렬합니다.

```
$ 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"
  }
]
```

자세한 내용은 JMESPath 웹 사이트의 sort_by를 참조하세요.

### 고급 **--query** 예제

#### 특정 항목에서 정보를 추출하는 방법

다음 예제에서는 목록에서 특정 항목을 찾은 다음 해당 항목에서 정보를 추출하는 데 --query 파라미터를 사용합니다. 이 예제에서는 지정된 서비스 엔드포인트에 연결된 모든 AvailabilityZones을(를) 나열합니다. 지정된 ServiceDetails을 가진 ServiceName 목록에서 해당 항목을 추출한 다음, 선택한 항목에서 AvailabilityZones 필드를 출력합니다.

```
$ 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"
    ]
]
```

지정된 생성 날짜 이후 스냅샷을 표시하는 방법

다음 예제에서는 출력에 사용 가능한 필드를 몇 개만 포함하여 지정된 날짜 이후에 생성된 모든 스냅샷을 나열하는 방법을 보여줍니다.

```
$ 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
    }
]
```

가장 최근 AMIs

다음 예제에서는 가장 최근 이미지에서 가장 오래된 이미지로 정렬하여 생성한 5개의 최신 Amazon Machine Images(AMIs)를 나열합니다.

```
$ 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"
    }
]
```

비정상 Auto Scaling 인스턴스를 표시하려면

다음 예제에서는 지정된 AutoScaling 그룹의 비정상 인스턴스에 대한 `InstanceId`만 보여줍니다.

```
$ aws autoscaling describe-auto-scaling-groups \
    --auto-scaling-group-name My-AutoScaling-Group-Name \
    --output text \
    --query 'AutoScalingGroups[*].Instances[?HealthStatus==`Unhealthy`].InstanceId'
```

지정된 태그가 있는 볼륨을 포함하는 방법

다음 예제에서는 `test` 태그가 있는 모든 인스턴스에 대해 설명합니다. 볼륨에 연결된 `test` 옆에 또 다른 태그가 있으면, 볼륨은 여전히 결과에 반환됩니다.

아래 표현식은 `test` 태그가 있는 모든 태그를 배열에 반환합니다. `test` 태그가 아닌 모든 태그에는 `null` 값이 포함됩니다.

```
$ aws ec2 describe-volumes \
    --query 'Volumes[*].Tags[?Value == `test`]'
```

지정된 태그가 있는 볼륨을 제외하는 방법

다음 예제에서는 `test` 태그가 없는 모든 인스턴스에 대해 설명합니다. 볼륨에 여러 태그가 있을 수 있으므로 단순 `?Value != `test`` 표현식을 사용하면 볼륨을 제외하지 않습니다. 볼륨에 연결된 `test` 옆에 또 다른 태그가 있으면, 볼륨은 여전히 결과에 반환됩니다.

`test` 태그가 있는 모든 볼륨을 제외하려면 아래 표현식으로 시작하여 `test` 태그가 있는 모든 태그를 배열에 반환합니다. `test` 태그가 아닌 모든 태그에는 `null` 값이 포함됩니다.

```
$ aws ec2 describe-volumes \
    --query 'Volumes[*].Tags[?Value == `test`]'
```

그런 다음 test 함수를 사용하여 모든 양의 not_null 결과를 필터링합니다.

```
$ aws ec2 describe-volumes \
    --query 'Volumes[!not_null(Tags[?Value == `test`].Value)]'
```

결과를 파이핑하여 결과를 병합하면 다음 쿼리가 생성됩니다.

```
$ aws ec2 describe-volumes \
    --query 'Volumes[!not_null(Tags[?Value == `test`].Value)] | []'
```

## 서버 측 필터링과 클라이언트 측 필터링 결합

서버 측 필터링과 클라이언트 측 필터링을 함께 사용할 수 있습니다. 서버 측 필터링이 먼저 완료되어, --query 매개 변수가 필터링하는 데이터를 클라이언트로 보냅니다. 대규모 데이터 세트를 사용하는 경우 먼저 서버 측 필터링을 사용하면 클라이언트 측 필터링이 제공하는 강력한 사용자 지정을 유지하면서 각 AWS CLI 호출에 대해 클라이언트로 전송되는 데이터의 양을 줄일 수 있습니다.

다음 예제에서는 서버 측 필터링과 클라이언트 측 필터링을 모두 사용하여 Amazon EC2 볼륨을 나열합니다. 이 서비스는 us-west-2a 가용 영역에서 연결된 모든 볼륨의 목록을 필터링합니다. --query 파라미터는 또한 출력을 50보다 큰 Size 값을 가진 볼륨으로만 제한하며 사용자 정의 이름으로 지정된 필드만 표시합니다.

```
$ 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"
    }
]
```

다음 예제에서는 여러 기준을 충족하는 이미지의 목록을 가져옵니다. 그런 다음 --query 파라미터를 사용하여 CreationDate를 기준으로 출력을 정렬하고 가장 최근 항목만 선택합니다. 마지막으로 해당 이미지의 ImageId를 표시합니다.

```
$ 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
```

다음 예제에서는 를 사용하여 목록에 있는 수를 `length` 계산IOPS하여 1,000개 이상의 사용 가능한 볼륨 수를 표시합니다.

```
$ aws ec2 describe-volumes \
    --filters "Name=status,Values=available" \
    --query 'length(Volumes[?Iops > `1000`])'
3
```

### 추가 리소스

JMESPath 터미널

JMESPath 터미널은 클라이언트 측 필터링에 사용되는 JMESPath 표현식을 실험하는 대화형 터미널 명령입니다. `jpterm` 명령을 사용하면 사용자가 입력할 때 터미널에 즉시 쿼리 결과가 표시됩니다. AWS CLI 출력을 터미널로 직접 파이프하여 고급 쿼리 실험을 활성화할 수 있습니다.

다음 예제는 `aws ec2 describe-volumes` 출력을 JMESPath 터미널로 직접 파이프합니다.

```
$ aws ec2 describe-volumes | jpterm
```

JMESPath 터미널 및 설치 지침에 대한 자세한 내용은 의 JMESPath 터미널을 참조하세요GitHub.

jq 유틸리티

`jq` 유틸리티는 클라이언트 측의 출력을 사용자가 원하는 출력 형식으로 변환하는 방법을 제공합니다. `jq` 및 설치 지침에 대한 자세한 내용은 의 jq를 참조하세요GitHub.

## 의 명령줄 반환 코드 AWS CLI

반환 코드는 일반적으로 명령의 상태를 설명하는 AWS Command Line Interface (AWS CLI) 명령을 실행한 후 전송되는 숨겨진 코드입니다. `echo` 명령을 사용하여 마지막 AWS CLI 명령에서 전송된 코드

를 표시하고 이러한 코드를 사용하여 명령이 성공했는지 또는 실패했는지, 명령에 오류가 발생한 이유를 확인할 수 있습니다. 반환 코드 외에도 `--debug` 스위치로 명령을 실행하여 실패에 대한 자세한 정보를 볼 수 있습니다. 이렇게 하면 AWS CLI 가 명령을 처리하기 위해 사용하는 단계와 각 단계의 결과가 포함된 세부 보고서가 생성됩니다.

AWS CLI 명령의 반환 코드를 확인하려면 명령을 실행한 직후 다음 CLI 명령 중 하나를 실행합니다.

Linux and macOS

```
$ echo $?
0
```

Windows PowerShell

```
PS> echo $lastexitcode
0
```

Windows Command Prompt

```
C:\> echo %errorlevel%
0
```

다음은 AWS Command Line Interface (AWS CLI) 명령을 실행할 때 반환할 수 있는 반환 코드 값입니다.

| 코드 | 의미 |
| --- | --- |
| 0 | 서비스는 HTTP 응답 상태 코드 200으로 응답했으며, 이는 요청이 전송된 AWS CLI 및 AWS 서비스에서 생성된 오류가 없음을 나타냅니다. |
| 1 | 하나 이상의 Amazon S3 전송 작업이 실패했습니다. S3 명령으로 제한됩니다. |
| 2 | 이 반환 코드의 의미는 명령에 따라 달라집니다.<br>• 모든 AWS CLI 명령에 적용 가능 - 입력한 명령을 구문 분석할 수 없습니다. 필수적인 하위 명령 또는 인수가 누락되거나 알려지지 않은 명령 또는 인수를 사용한 것이 구문 분석이 실패한 이유 중 하나일 수 있습니다. |

| 코드 | 의미 |
| --- | --- |
| | • S3 명령으로 제한됨 - 전송 프로세스에서 전송 대상으로 표시된 파일을 하나 이상 건너 뛰었습니다. 그러나 이전 대상으로 표시된 다른 모든 파일들은 성공적으로 전송되었습니다. 전송 프로세스 중에 건너뛴 파일에는 존재하지 않는 파일, 특수 디바이스 문자, 특수 디바이스 차단, FIFO 대기열 또는 소켓인 파일, 사용자에게 읽기 권한이 없는 파일이 포함됩니다. |
| 130 | 명령이 에 의해 중단되었습니다SIGINT. 이것은 `Ctrl+C`로 명령을 취소하기 위해 사용자가 보낸 신호입니다. |
| 255 | 명령이 실패했습니다. AWS CLI 또는 요청이 전송된 서비스에서 AWS 오류가 발생했습니다. |

# 에서 별칭 생성 및 사용 AWS CLI

별칭은 자주 사용하는 명령 또는 스크립트를 단축하기 위해 AWS Command Line Interface (AWS CLI)에서 생성할 수 있는 바로 가기입니다. 구성 폴더에 있는 `alias` 파일에서 별칭을 생성합니다.

주제



## 사전 조건

별칭 명령을 사용하려면 다음을 완료해야 합니다.

- AWS CLI를 설치하고 구성합니다. 자세한 내용은 설치 AWS CLI 및 에 대한 인증 및 액세스 자격 증명 AWS CLI 단원을 참조하세요.
- 1.11.24 또는 2.0.0의 최소 AWS CLI 버전을 사용합니다.
- (선택 사항) AWS CLI 별칭 bash 스크립트를 사용하려면 bash 호환 터미널을 사용해야 합니다.

# 1단계: 별칭 파일 생성

`alias` 파일을 생성하려면 파일 탐색 및 텍스트 편집기를 사용하거나 절차를 사용하여 원하는 터미널을 step-by-step 사용할 수 있습니다. 별칭 파일을 빠르게 생성하려면 다음 명령 블록을 사용합니다.

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
```

별칭 파일을 생성하는 방법

1. AWS CLI 구성 폴더에 `cli` 라는 폴더를 생성합니다. 기본적으로 구성 폴더는 `~/.aws/`(Linux 또는 macOS) 및 `%USERPROFILE%\.aws\`(Windows)에 있습니다. 파일 탐색을 통해 또는 다음 명령을 사용하여 생성할 수 있습니다.

   Linux and macOS

   ```
   $ mkdir -p ~/.aws/cli
   ```

   Windows

   ```
   C:\> md %USERPROFILE%\.aws\cli
   ```

   결과로 생성되는 `cli` 폴더의 기본 경로는 `~/.aws/cli/`(Linux 또는 macOS) 및 `%USERPROFILE%\.aws\cli`(Windows)입니다.

2. `cli` 폴더에서 확장자 없이 이름이 `alias`인 텍스트 파일을 생성하고 첫 번째 줄에 `[toplevel]`을 추가합니다. 원하는 텍스트 편집기를 통해 또는 다음 명령을 사용하여 이 파일을 생성할 수 있습니다.

Linux and macOS

```
$ echo '[toplevel]' > ~/.aws/cli/alias
```

Windows

```
C:\> echo [toplevel] > %USERPROFILE%/.aws/cli/alias
```

## 2단계: 별칭 생성

기본 명령이나 bash 스크립팅을 사용하여 별칭을 생성할 수 있습니다.

### 기본 명령 별명 생성

이전 단계에서 생성한 `alias` 파일에서 다음 구문을 사용해 명령을 추가하여 별칭을 생성할 수 있습니다.

구문

```
aliasname = command [--options]
```

는 *aliasname* 는 별칭이라고 합니다. 는 *command* 는 호출하려는 명령으로, 다른 별칭을 포함할 수 있습니다. 별칭에 옵션 또는 파라미터를 포함하거나 별칭을 호출할 때 추가할 수 있습니다.

다음 예제에서는 `aws sts get-caller-identity` 명령을 사용하여 이름이 `aws whoami`인 별칭을 생성합니다. 이 별칭은 기존 AWS CLI 명령을 호출하므로 `aws` 접두사 없이 명령을 작성할 수 있습니다.

```
whoami = sts get-caller-identity
```

다음 예제에서는 이전 `whoami` 예제를 사용하여 `Account` 필터 및 텍스트 `output` 옵션을 추가합니다.

```
whoami2 = sts get-caller-identity --query Account --output text
```

## 하위 명령 별칭 만들기

> **Note**
>
> 하위 명령 별칭 기능에는 1.11.24 또는 2.0.0의 최소 AWS CLI 버전이 필요합니다.

이전 단계에서 생성한 `alias` 파일에서 다음 구문을 사용해 명령을 추가하여 하위 명령 별칭을 생성할 수 있습니다.

구문

```
[command commandGroup]
aliasname = command [--options]
```

는 *commandGroup* 는 명령 네임스페이스입니다. 예를 들어 명령`aws ec2 describe-regions`은 `ec2` 명령 그룹 아래에 있습니다. 는 *aliasname* 는 별칭이라고 합니다. 는 *command* 는 호출하려는 명령으로, 다른 별칭을 포함할 수 있습니다. 별칭에 옵션 또는 파라미터를 포함하거나 별칭을 호출할 때 추가할 수 있습니다.

다음 예제에서는 `aws ec2 describe-regions` 명령을 사용하여 이름이 `aws ec2 regions`인 별칭을 생성합니다. 이 별칭은 `ec2` 명령 네임스페이스 아래의 기존 AWS CLI 명령을 호출하므로 `aws ec2` 접두사 없이 명령을 작성할 수 있습니다.

```
[command ec2]
regions = describe-regions --query Regions[].RegionName
```

명령 네임스페이스 외부의 명령에서 별칭을 만들려면 전체 명령 앞에 느낌표를 붙입니다. 다음 예제에서는 `aws iam list-instance-profiles` 명령을 사용하여 이름이 `aws ec2 instance-profiles`인 별칭을 생성합니다.

```
[command ec2]
instance-profiles = !aws iam list-instance-profiles
```

> **Note**
>
> 별칭은 기존 명령 네임스페이스만 사용하며 새 명령 네임스페이스를 만들 수 없습니다. 예를 들어 `johnsmith` 명령 네임스페이스가 이미 존재하지 않으므로 `[command johnsmith]` 섹션을 사용하여 별칭을 만들 수 없습니다.

## bash 스크립팅 별칭 생성

> **Warning**
>
> AWS CLI 별칭 bash 스크립트를 사용하려면 bash 호환 터미널을 사용해야 합니다.

다음 구문을 사용하여 고급 프로세스에 대한 bash 스크립트를 사용하여 별칭을 생성할 수 있습니다.

구문

```
aliasname =
    !f() {
        script content
}; f
```

는 *aliasname* 는 별칭을 호출하며 *script content* 는 별칭을 호출할 때 실행하려는 스크립트입니다.

다음 예제에서는 opendns를 사용하여 현재 IP 주소를 출력합니다. 다른 별칭에서 별칭을 사용할 수 있으므로 다음 `myip` 별칭은 다른 별칭 내 IP 주소에 대한 액세스를 허용하거나 취소하는 데 유용합니다.

```
myip =
  !f() {
    dig +short myip.opendns.com @resolver1.opendns.com
  }; f
```

다음 스크립트 예제에서는 이전 `aws myip`별칭을 호출하여 Amazon EC2 보안 그룹 수신에 대한 IP 주소를 승인합니다.

```
authorize-my-ip =
  !f() {
```

```
    ip=$(aws myip)
    aws ec2 authorize-security-group-ingress --group-id ${1} --cidr $ip/32 --protocol
 tcp --port 22
  }; f
```

bash 스크립팅을 사용하는 별칭을 호출하면 변수는 항상 입력한 순서대로 전달됩니다. bash 스크립팅에서 변수 이름은 고려하지 않고 나타나는 순서만 고려합니다. 다음 `textalert` 별칭 예제에서 `--message` 옵션에 대한 변수는 첫 번째이고 `--phone-number` 옵션은 두 번째입니다.

```
textalert =
  !f() {
    aws sns publish --message "${1}" --phone-number ${2}
  }; f
```

## 3단계: 별칭 호출

`alias` 파일에서 생성한 별칭을 실행하려면 다음 구문을 사용합니다. 별칭을 호출할 때 추가 옵션을 추가할 수 있습니다.

구문

```
$ aws aliasname
```

다음 예제에서는 `aws whoami` 명령 별칭을 사용합니다.

```
$ aws
 whoami
{
    "UserId": "A12BCD34E5FGHI6JKLM",
    "Account": "1234567890987",
    "Arn": "arn:aws:iam::1234567890987:user/userName"
}
```

다음 예제에서는 `aws whoami` 별칭을 추가 옵션과 함께 사용하여 `Account` 출력에서 `text` 번호만 반환합니다.

```
$ aws whoami --query Account --output
 text
1234567890987
```

다음 예제에서는 `aws ec2 regions` 하위 명령 별칭을 사용합니다.

```
$ aws ec2
 regions
[
    "ap-south-1",
    "eu-north-1",
    "eu-west-3",
    "eu-west-2",
...
```

## bash 스크립팅 변수를 사용하여 별칭 호출

bash 스크립팅을 사용하는 별칭을 호출하면 변수는 입력한 순서대로 전달됩니다. bash 스크립팅에서 변수 이름은 고려하지 않고 나타나는 순서만 고려합니다. 예를 들어, 다음 `textalert` 별칭에서 `--message` 옵션에 대한 변수는 첫 번째이고 `--phone-number`는 두 번째입니다.

```
textalert =
  !f() {
    aws sns publish --message "${1}" --phone-number ${2}
  }; f
```

`textalert` 별칭을 호출할 때 별칭에서 실행되는 동일한 순서로 변수를 전달해야 합니다. 다음 예제에서는 $message 및 $phone 변수를 사용합니다. $message 변수는 ${1} 옵션에서 `--message`(으)로 전달되고, $phone 변수는 ${2} 옵션에서 `--phone-number`(으)로 전달됩니다. 이렇게 하면 `textalert` 별칭을 호출하여 메시지를 보낼 수 있습니다.

```
$ aws textalert $message
 $phone
{
    "MessageId": "1ab2cd3e4-fg56-7h89-i01j-2klmn34567"
}
```

다음 예제에서는 $phone 및 $message에 대한 별칭을 호출할 때 순서가 전환됩니다. $phone 변수는 ${1} 옵션에서 `--message`(으)로 전달되고, $message 변수는 ${2} 옵션에서 `--phone-number`(으)로 전달됩니다. 변수 순서가 잘못되었으므로 별칭이 변수를 잘못 전달합니다. 이 경우 $message 콘텐츠가 `--phone-number` 옵션의 전화번호 형식 요구 사항과 일치하지 않으므로 오류가 발생합니다.

```
$ 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
```

## 별칭 리포지토리 예제

의 AWS CLI 별칭 리포지토리에는 AWS CLI 개발자 팀과 커뮤니티에서 생성한 별 AWS CLI 칭 예제가 GitHub 포함되어 있습니다. 전체 `alias` 파일 예제를 사용하거나 직접 사용할 개별 별칭을 사용할 수 있습니다.

> **Warning**
>
> 이 섹션의 명령을 실행하면 기존 `alias` 파일이 삭제됩니다. 기존 별칭 파일을 덮어쓰지 않으려면 다운로드 위치를 변경합니다.

리포지토리에서 별칭을 사용하는 방법

1. Git를 설치합니다. 설치 지침은 Git 설명서에서 Getting Started - Installing Git를 참조하세요.
2. `jp` 명령을 설치합니다. `jp` 명령은 `tostring` 별칭에 사용됩니다. 설치 지침은 의 JMESPath (jp) README.md를 참조하세요GitHub.
3. `jq` 명령을 설치합니다. `jq` 명령은 `tostring-with-jq` 별칭에 사용됩니다. 설치 지침은 의 JSON 프로세서(jq)를 참조하세요GitHub.
4. 다음 중 하나를 수행하여 `alias` 파일을 다운로드합니다.

   - 리포지토리에서 다운로드한 다음 명령을 실행하여 `alias` 파일을 구성 폴더에 복사합니다.

     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
```

- 리포지토리에서 직접 다운로드하고 AWS CLI 구성 `cli` 폴더의 폴더에 저장합니다. 기본적으로 구성 폴더는 `~/.aws/`(Linux 또는 macOS) 및 `%USERPROFILE%\.aws\`(Windows)에 있습니다.

5. 별칭이 작동하는지 확인하려면 다음 별칭을 실행합니다.

```
$ aws whoami
```

그러면 `aws sts get-caller-identity` 명령과 동일한 응답이 표시됩니다.

```
{
    "Account": "012345678901",
    "UserId": "AIUAINBADX2VEG2TC6HD6",
    "Arn": "arn:aws:iam::012345678901:user/myuser"
}
```

## 리소스

- 의 AWS CLI 별칭 리포지토리에는 AWS CLI 개발자 팀이 생성한 별 AWS CLI 칭 예제와 커뮤니티의 기여가 GitHub 포함되어 있습니다 AWS CLI .
- AWS re:Invent 2016의 별칭 기능 공지: 의 유효 AWS CLI 사용자YouTube.
- `aws sts get-caller-identity`
- `aws ec2 describe-instances`
- `aws sns publish`

# 에 대한 코드 예제 AWS CLI

이 장에서는 에서 (AWS CLI)를 AWS Command Line Interface 사용하는 방법을 보여주는 예제 모음을 제공합니다 AWS 서비스.

AWS CLI 에는 이 안내서의 다음과 같은 유형의 예제가 있습니다.

- 가이드 명령 예제 - 일부 에서 를 사용하는 방법에 대한 AWS CLI 사용 설명서의 가이드 명령 예제 AWS CLI 입니다 AWS 서비스. 이는 AWS CLI 참조 가이드 의 예제보다 더 자세한 예제인 경우가 많습니다.
- AWS CLI 명령 예제 - AWS CLI 참조 가이드 . 명령 예제는 의 AWS CLI리포지토리에서 호스팅됩니다GitHub.
- AWS CLI Bash 스크립팅 코드 예제 사용 - 오픈 소스 bash 스크립팅 예제. Bash 스크립팅 예제는 의 AWS 코드 예제 리포지토리에서 호스팅됩니다GitHub.

> 예제 피드백
>
> 필요한 예제를 찾을 수 없습니까? 이 페이지 하단의 피드백 제공 링크 또는 AWS CLI 참조 가이드 의 관련 명령 페이지를 사용하여 명령 예제를 요청합니다.
> 기여하고 싶으십니까? 의 AWS 코드 예제 리포지토리에 AWS CLI 명령 예제를 기여합니다GitHub. 기여에 대한 자세한 내용은 GitHub 페이지의 AWS CLI 코드 예제 기여 빠른 단계를 참조하세요.

# 에 대한 가이드 명령 예제 AWS CLI

AWS Command Line Interface (AWS CLI)는 명령줄 쉘에서 다양한 명령을 AWS 서비스 사용하여 상호 작용할 수 있는 오픈 소스 도구입니다. 이 섹션에서는 를 활용하여 일부 AWS CLI 에 액세스하는 방법을 보여주는 가이드 예제를 제공합니다 AWS 서비스. 여기에는 상위 수준 AWS CLI 명령과 같은 일부 사용자 지정 `aws s3` 명령이 포함됩니다. 이 명령 예제는 일부 에 사용되는 일반적인 작업을 보여주고 자세한 정보를 위한 추가 리소스를 AWS 서비스 제공합니다.

경험이 풍부한 AWS 사용자이든 를 처음 사용하는 사용자이든 AWS CLI이 가이드 예제는 AWS 작업을 간소화하기 위한 리소스 역할을 합니다.

각 에 사용 가능한 모든 명령에 대한 전체 참조는 AWS CLI 참조 가이드 를 AWS 서비스참조하세요. 또한 기본 제공 명령줄 도움말을 사용하여 에서 AWS 서비스, 명령, 옵션 및 기능의 배열을 탐색할 수 있습니다 AWS CLI.

이 섹션에서 사용할 수 없는 자세한 명령 예제는 AWS CLI 명령 예제 섹션을 참조하세요. 다음은 AWS CLI 참조 가이드 에서도 사용할 수 있는 오픈 소스 명령 예제입니다. 명령 예제는 의 리AWS CLI포지토리에서 호스팅됩니다GitHub.

오픈 소스 bash 스크립팅 예제는 섹션을 참조하세요the section called “Bash 스크립트 예제”. Bash 스크립팅 예제는 의 AWS 코드 예제 리포지토리에서 호스팅됩니다GitHub.

서비스

- 에서 Amazon DynamoDB 사용 AWS CLI
- EC2 에서 Amazon 사용 AWS CLI
- 에서 Amazon S3 Glacier 사용 AWS CLI
- IAM 에서 사용 AWS CLI
- 에서 Amazon S3 사용 AWS CLI
- SNS 에서 Amazon 액세스 AWS CLI

## 에서 Amazon DynamoDB 사용 AWS CLI

| Amazon DynamoDB 소개 |
| --- |
| What is Amazon DynamoDB? |

AWS Command Line Interface (AWS CLI)는 Amazon DynamoDB를 포함한 모든 AWS 데이터베이스 서비스를 지원합니다. 테이블 생성과 같은 즉석 작업에 AWS CLI 를 사용할 수 있습니다. 또한 이를 사용하여 DynamoDB 작업을 유틸리티 스크립트 내에 포함할 수 있습니다.

DynamoDB와 AWS CLI 함께 를 사용하는 방법에 대한 자세한 내용은 명령 참조dynamodb의 섹션을 참조하세요. AWS CLI

DynamoDB 에 대한 AWS CLI 명령을 나열하려면 다음 명령을 사용합니다.

```
$ aws dynamodb help
```

주제



## 사전 조건

dynamodb 명령을 실행하려면 다음을 수행해야 합니다.

- AWS CLI를 설치하고 구성합니다. 자세한 내용은 설치 AWS CLI 및 에 대한 인증 및 액세스 자격 증명 AWS CLI 단원을 참조하세요.

## DynamoDB 테이블 생성 및 사용

명령줄 형식은 DynamoDB 명령 이름과 해당 명령에 대한 파라미터 순서로 구성됩니다. 는 파라미터 값 및 전체 에 대한 CLI 단축 구문을 AWS CLI 지원합니다JSON. ???

다음 예제에서는 MusicCollection이라는 테이블을 생성합니다.

```
$ 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
```

다음 예제와 유사한 명령을 사용하여 새 줄을 테이블에 추가할 수 있습니다. 이 예제에서는 단축형 구문과 의 조합을 사용합니다JSON.

```
$ 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"
    }
}
```

단일 라인 명령JSON에서 유효한 를 구성하는 것은 어려울 수 있습니다. 이를 더 쉽게 하기 위해 는 JSON 파일을 읽을 AWS CLI 수 있습니다. 예를 들어 라는 파일에 저장되는 다음 JSON 조각을 고려해 보세요`expression-attributes.json`.

```
{
  ":v1": {"S": "No One You Know"},
  ":v2": {"S": "Call Me Today"}
}
```

이 파일을 사용하면 `query`를 사용하여 AWS CLI요청을 발행할 수 있습니다. 다음 예제에서는 `expression-attributes.json` 파라미터의 값으로 `--expression-attribute-values` 파일의 콘텐츠가 사용됩니다.

```
$ 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
}
```

## DynamoDB Local 사용

DynamoDB 외에도 DynamoDB Local과 AWS CLI 함께 를 사용할 수 있습니다. DynamoDB Local은 DynamoDB 서비스를 모방하는 클라이언트 측 소형 데이터베이스 및 서버입니다. DynamoDB Local을 사용하면 DynamoDB 웹 서비스에서 테이블 또는 데이터를 조작API하지 않고도 DynamoDB를 사용하는 애플리케이션을 작성할 수 있습니다. 대신 모든 API 작업이 로컬 데이터베이스로 다시 라우팅됩니다. 이를 통해 프로비저닝된 처리량, 데이터 스토리지 및 데이터 전송 요금을 절감할 수 있습니다.

DynamoDB Local에 대한 자세한 내용과 에서 DynamoDB Local을 사용하는 방법은 Amazon DynamoDB 개발자 안내서의 다음 섹션을 AWS CLI참조하세요. DynamoDB

- DynamoDB Local
- DynamoDB Local과 함께 AWS CLI 사용

## 리소스

AWS CLI 참조:

- `aws dynamodb`
- `aws dynamodb create-table`
- `aws dynamodb put-item`
- `aws dynamodb query`

서비스 참조:

- Amazon DynamoDB 개발자 안내서의 DynamoDB Local
- Amazon DynamoDB 개발자 안내서의 AWS CLI 에서 DynamoDB Local 사용

# EC2 에서 Amazon 사용 AWS CLI

| Amazon Elastic Compute Cloud 소개 |
| --- |
| Amazon EC2- Elastic Cloud Server 및 를 사용한 호스팅 소개 AWS |

Amazon Elastic Compute Cloud(Amazon EC2)는 확장성과 유연성이 뛰어난 가상 컴퓨팅 환경을 제공합니다. Amazon을 EC2 사용하면 Amazon EC2 인스턴스라고 하는 가상 서버를 프로비저닝하고 관리하여 광범위한 컴퓨팅 요구 사항을 충족할 수 있습니다.

Amazon EC2 인스턴스는 CPU, 메모리, 스토리지 및 네트워킹 기능의 다양한 구성으로 사용자 지정할 수 있는 가상 머신입니다. 애플리케이션 요구 사항에 따라 가볍고 비용 효율적인 옵션부터 강력한 고성능 인스턴스에 이르기까지 다양한 인스턴스 유형 중에서 선택할 수 있습니다. 이러한 유연성을 통해 컴퓨팅 요구 사항에 맞게 성능과 비용 효율성을 최적화할 수 있습니다.

또한 AmazonEC2은 컴퓨팅 리소스를 효과적으로 관리할 수 있는 기능을 제공합니다. 여기에는 새 인스턴스를 빠르게 시작하고, 빠른 배포를 위해 사용자 지정 기계 이미지(AMIs)를 생성하고, 필요에 따라 컴퓨팅 용량을 늘리거나 줄이는 기능이 포함됩니다.

AWS Command Line Interface ()를 EC2 사용하여 Amazon의 기능에 액세스할 수 있습니다AWS CLI. Amazon 에 대한 AWS CLI 명령을 나열하려면 다음 명령을 EC2사용합니다.

```
aws ec2 help
```

명령을 실행하기 전에 기본 자격 증명을 설정합니다. 자세한 내용은 에 대한 설정 구성 AWS CLI 단원을 참조하십시오.

이 주제에서는 Amazon 에 대한 일반적인 작업을 수행하는 AWS CLI 명령의 간단한 예제를 보여줍니다EC2.

AWS CLI 명령의 긴 형식 예제는 의 AWS CLI 코드 예제 리포지토리를 참조하세요GitHub.

주제

- 에서 Amazon EC2 키 페어 생성, 표시 및 삭제 AWS CLI
- 에서 Amazon EC2 보안 그룹 생성, 구성 및 삭제 AWS CLI
- 에서 Amazon EC2 인스턴스 시작, 나열 및 종료 AWS CLI
- 에서 bash 스크립트로 Amazon EC2 인스턴스 유형 변경 AWS CLI

## 에서 Amazon EC2 키 페어 생성, 표시 및 삭제 AWS CLI

AWS Command Line Interface (AWS CLI)를 사용하여 Amazon Elastic Compute Cloud(Amazon)에 대한 키 페어를 생성, 표시 및 삭제할 수 있습니다EC2. 키 페어를 사용하여 Amazon EC2 인스턴스에 연결합니다.

인스턴스를 생성할 EC2 때 Amazon에 키 페어를 제공한 다음 해당 키 페어를 사용하여 인스턴스에 연결할 때 인증해야 합니다.

> **Note**
>
> 추가 명령 예제는 AWS CLI 참조 가이드 를 참조하세요.

주제

- 사전 조건
- 키 페어 생성
- 키 페어 표시
- 키 페어 삭제
- 참조

사전 조건

ec2 명령을 실행하려면 다음을 수행해야 합니다.

- AWS CLI를 설치하고 구성합니다. 자세한 내용은 설치 AWS CLI 및 에 대한 인증 및 액세스 자격 증명 AWS CLI 단원을 참조하세요.
- Amazon EC2 액세스를 허용하도록 IAM 권한을 설정합니다. Amazon 에 대한 IAM 권한에 대한 자세한 내용은 Amazon EC2 사용 설명서의 Amazon 정책을 EC2참조하세요. IAM EC2

키 페어 생성

키 페어를 생성하려면 aws ec2 create-key-pair 명령과 함께 --query 옵션 및 --output text 옵션을 사용하여 프라이빗 키를 직접 파일에 파이프합니다.

```
$ aws ec2 create-key-pair --key-name MyKeyPair --query 'KeyMaterial' --output text
 > MyKeyPair.pem
```

의 경우 PowerShell리> file디렉션은 기본적으로 UTF-8 인코딩으로 설정되며, 이는 일부 SSH 클라이언트에서 사용할 수 없습니다. 따라서 out-file 명령으로 파이프하여 출력을 변환하고, 명시적으로 인코딩을 ascii로 설정해야 합니다.

```
PS C:\>aws ec2 create-key-pair --key-name MyKeyPair --query 'KeyMaterial' --output text
 | out-file -encoding ascii -filepath MyKeyPair.pem
```

결과로 나온 MyKeyPair.pem 파일은 다음과 같습니다.

```
-----BEGIN RSA PRIVATE KEY-----
EXAMPLEKEYKCAQEAy7WZhaDsrA1W3mRlQtvhwyORRX8gnxgDAfRt/gx42kWXsT4rXE/b5CpSgie/
vBoU7jLxx92pNHoFnByP+Dc21eyyz6CvjTmWA0JwfWiW5/akH7iO5dSrvC7dQkW2duV5QuUdE0QW
Z/aNxMniGQE6XAgfwlnXVBwrerrQo+ZWQeqiUwwMkuEbLeJFLhMCvYURpUMSC1oehm449ilx9X1F
G50TCFeOzfl8dqqCP6GzbPaIjiU19xX/azOR9V+tpUOzEL+wmXnZt3/nHPQ5xvD2OJH67km6SuPW
oPzev/D8V+x4+bHthfSjR9Y7DvQFjfBVwHXigBdtZcU2/wei8D/HYwIDAQABAoIBAGZ1kaEvnrqu
/uler7vgIn5m7lN5LKw4hJLAIW6tUT/fzvtcHK0SkbQCQXuriHmQ2MQyJX/0kn2NfjLV/ufGxbL1
mb5qwMGUnEpJaZD6QSSs3kICLwWUYUiGfc0uiSbmJoap/GTLU0W5Mfcv36PaBUNy5p53V6G7hXb2
bahyWyJNfjLe4M86yd2YK3V2CmK+X/BOsShnJ36+hjrXPPWmV3N9zEmCdJjA+K15DYmhm/tJWSD9
81oGk9TopEp7CkIfatEATyyZiVqoRq6k64iuM9JkA3OzdXzMQexXVJ1TLZVEH0E7bhlY9d8O1ozR
oQs/FiZNAx2iijCWyv0lpjE73+kCgYEA9mZtyhkHkFDpwrSM1APaL8oNAbbjwEy7Z5Mqfql+lIp1
YkriL0DbLXlvRAH+yHPRit2hHOjtUNZh4Axv+cpg09qbUI3+43eEy24B7G/Uh+GTfbjsXsOxQx/x
p9otyVwc7hsQ5TA5PZb+mvkJ5OBEKzet9XcKwONBYELGhnEPe7cCgYEA06Vgov6YHleHui9kHuws
ayav0elc5zkxjF9nfHFJRry21R1trw2Vdpn+9g481URrpzWVOEihvm+xTtmaZlSp//lkq75XDwnU
WA8gkn6O3QE3fq2yN98BURsAKdJfJ5RL1HvGQvTe10HLYYXpJnEkHv+Unl2ajLivWUt5pbBrKbUC
gYBjbO+OZk0sCcpZ29sbzjYjpIddErySIyRX5gV2uNQwAjLdp9PfN295yQ+BxMBXiIycWVQiw0bH
oMo7yykABY7Ozd5wQewBQ4AdSlWSX4nGDtsiFxWiI5sKuAAeOCbTosy1s8w8fxoJ5Tz1sdoxNeGs
Arq6Wv/G16zQuAE9zK9vvwKBgF+09VI/1wJBirsDGz9whVWfFPrTkJNvJZzYt69qezxlsjgFKshy
WBhd4xHZtmCqpBPlAymEjr/TOlbxyARmXMnIOWIAnNXMGB4KGSyl1mzSVAoQ+fqR+cJ3d0dyPl1j
jjb0Ed/NY8frlNDxAVHE8BSkdsx2f6ELEyBKJSRr9snRAoGAMrTwYneXzvTskF/S5Fyu0iOegLDa
NWUH38v/nDCgEpIXD5Hn3qAEcju1IjmbwlvtW+nY2jVhv7UGd8MjwUTNGItdb6nsYqM2asrnF3qS
VRkAKKKYeGjkpUfVTrW0YFjXkfcrR/V+QFL5OndHAKJXjW7a4ejJLncTzmZSpYzwApc=
-----END RSA PRIVATE KEY-----
```

프라이빗 키는 에 저장되지 AWS 않으며 생성된 경우에만 검색할 수 있습니다. 나중에 복구할 수 없습니다. 대신, 프라이빗 키를 잃어버리면 새 키 페어를 생성해야 합니다.

Linux 컴퓨터에서 인스턴스에 연결하는 경우, 사용자만 프라이빗 키 파일을 읽을 수 있도록 다음 명령으로 해당 권한을 설정하는 것이 좋습니다.

```
$ chmod 400 MyKeyPair.pem
```

키 페어 표시

키 페어에서 "지문"이 생성되고 이 지문을 사용하여 로컬 시스템에 있는 프라이빗 키가 AWS에 저장된 퍼블릭 키와 일치하는지 확인할 수 있습니다.

지문은 프라이빗 키의 DER인코딩된 복사본에서 가져온 SHA1해시입니다. 이 값은 키 페어가 생성될 때 캡처되고 퍼블릭 키와 AWS 함께 에 저장됩니다. Amazon EC2 콘솔에서 또는 AWS CLI 명령을 실행하여 지문을 볼 수 있습니다aws ec2 describe-key-pairs.

다음 예제는 MyKeyPair의 지문을 표시합니다.

```
$ 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"
        }
    ]
}
```

키 및 지문에 대한 자세한 내용은 Amazon 사용 설명서의 Amazon EC2 키 페어를 참조하세요. EC2

키 페어 삭제

키 페어를 삭제하려면 aws ec2 delete-key-pair 명령을 실행하고 *MyKeyPair* 삭제할 페어의 이름이 표시됩니다.

```
$ aws ec2 delete-key-pair --key-name MyKeyPair
```

참조

AWS CLI 참조:

- aws ec2
- aws ec2 create-key-pair
- aws ec2 delete-key-pair
- aws ec2 describe-key-pairs

기타 참조:

- Amazon Elastic Compute Cloud 설명서
- 및 AWS CLI 코드 예제를 AWS SDK 보고 기여하려면 의 AWS 코드 예제 리포지토리를 참조하세요GitHub.

## 에서 Amazon EC2 보안 그룹 생성, 구성 및 삭제 AWS CLI

기본적으로 방화벽으로 작동하는 Amazon Elastic Compute Cloud(Amazon EC2) 인스턴스에 대한 보안 그룹을 생성할 수 있으며, 이 보안 그룹은 어떤 네트워크 트래픽이 들어오고 나갈 수 있는지 결정하는 규칙을 포함합니다.

AWS Command Line Interface (AWS CLI)를 사용하여 보안 그룹을 생성하고, 기존 보안 그룹에 규칙을 추가하고, 보안 그룹을 삭제합니다.

> **Note**
>
> 추가 명령 예제는 AWS CLI 참조 가이드 를 참조하세요.

주제

- 사전 조건
- 보안 그룹 생성
- 보안 그룹에 규칙 추가
- 보안 그룹 삭제
- 참조

사전 조건

ec2 명령을 실행하려면 다음을 수행해야 합니다.

- AWS CLI를 설치하고 구성합니다. 자세한 내용은 설치 AWS CLI 및 에 대한 인증 및 액세스 자격 증명 AWS CLI 단원을 참조하세요.
- Amazon EC2 액세스를 허용하도록 IAM 권한을 설정합니다. Amazon 에 대한 IAM 권한에 대한 자세한 내용은 Amazon EC2 사용 설명서의 Amazon 정책을 EC2참조하세요. IAM EC2

보안 그룹 생성

가상 프라이빗 클라우드(VPCs)와 연결된 보안 그룹을 생성할 수 있습니다.

다음 aws ec2 create-security-group 예제에서는 지정된 에 대한 보안 그룹을 생성하는 방법을 보여줍니다VPC.

```
$ aws ec2 create-security-group --group-name my-sg --description "My security group" --
vpc-id vpc-1a2b3c4d
{
    "GroupId": "sg-903004f8"
}
```

보안 그룹에 대한 초기 정보를 보려면 aws ec2 describe-security-groups 명령을 실행합니다. EC2-VPC 보안 그룹은 이름이 vpc-id아닌 로만 참조할 수 있습니다.

```
$ 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"
        }
    ]
}
```

보안 그룹에 규칙 추가

Amazon EC2 인스턴스를 실행할 때 보안 그룹에서 규칙을 활성화하여 이미지에 연결하는 수단에 대한 수신 네트워크 트래픽을 허용해야 합니다.

예를 들어 Windows 인스턴스를 시작하는 경우 일반적으로 TCP 포트 3389의 인바운드 트래픽이 원격 데스크톱 프로토콜()을 지원하도록 허용하는 규칙을 추가합니다RDP. Linux 인스턴스를 시작하는 경우 일반적으로 TCP 포트 22의 인바운드 트래픽이 SSH 연결을 지원하도록 허용하는 규칙을 추가합니다.

`aws ec2 authorize-security-group-ingress` 명령을 사용하여 보안 그룹에 규칙을 추가합니다. 이 명령의 필수 파라미터는 컴퓨터의 퍼블릭 IP 주소 또는 컴퓨터가 연결된 네트워크(주소 범위 형식)CIDR입니다.

> **Note**
>
> 퍼블릭 IP 주소를 확인할 수 있도록 https://checkip.amazonaws.com/ 서비스를 제공합니다. IP 주소를 식별하는 데 도움이 되는 다른 서비스를 찾으려면 브라우저를 사용하여 "내 IP 주소"를 검색합니다. 동적 IP 주소(사설 네트워크의 NAT 게이트웨이를 통해)를 사용하여 방화벽 뒤에서 ISP 또는 를 통해 연결하는 경우 주소가 주기적으로 변경될 수 있습니다. 이 경우 클라이언트 컴퓨터에서 사용하는 IP 주소 범위를 알아야 합니다.

다음 예제에서는 IP 주소를 `sg-903004f8` 사용하여 ID가 인 EC2-VPC 보안 그룹에 RDP (TCP 포트 3389)에 대한 규칙을 추가하는 방법을 보여줍니다.

시작하려면 IP 주소를 찾습니다.

```
$ curl https://checkip.amazonaws.com
x.x.x.x
```

그런 다음 `aws ec2 authorize-security-group-ingress` 명령을 실행하여 IP 주소를 보안 그룹에 추가할 수 있습니다.

```
$ aws ec2 authorize-security-group-ingress --group-id sg-903004f8 --protocol tcp --port
 3389 --cidr x.x.x.x/x
```

다음 명령은 동일한 보안 그룹의 인스턴스에 활성화SSH할 또 다른 규칙을 추가합니다.

```
$ aws ec2 authorize-security-group-ingress --group-id sg-903004f8 --protocol tcp --port
 22 --cidr x.x.x.x/x
```

보안 그룹의 변경 사항을 보려면 aws ec2 describe-security-groups 명령을 실행합니다.

```
$ 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"
        }
    ]
}
```

보안 그룹 삭제

보안 그룹을 삭제하려면 aws ec2 delete-security-group 명령을 실행합니다.

> **Note**
> 환경에 현재 연결되어 있는 경우 보안 그룹을 삭제할 수 없습니다.

다음 명령 예제에서는 EC2-VPC 보안 그룹을 삭제합니다.

```
$ aws ec2 delete-security-group --group-id sg-903004f8
```

참조

AWS CLI 참조:

- aws ec2
- aws ec2 authorize-security-group-ingress
- aws ec2 create-security-group
- aws ec2 delete-security-group
- aws ec2 describe-security-groups

기타 참조:

- Amazon Elastic Compute Cloud 설명서
- 및 AWS CLI 코드 예제를 AWS SDK 보고 기여하려면 의 AWS 코드 예제 리포지토리를 참조하세요GitHub.

## 에서 Amazon EC2 인스턴스 시작, 나열 및 종료 AWS CLI

AWS Command Line Interface (AWS CLI)를 사용하여 Amazon Elastic Compute Cloud(AmazonEC2) 인스턴스를 시작, 나열 및 종료할 수 있습니다. AWS 프리 티어 내에 있지 않은 인스턴스를 시작하는 경우 인스턴스를 시작한 후 요금이 청구되고 유휴 상태로 유지되더라도 인스턴스가 실행 중인 시간에 대한 요금이 청구됩니다.

> **Note**
>
> 추가 명령 예제는 AWS CLI 참조 가이드 를 참조하세요.

주제



## 사전 조건

이 주제의 `ec2` 명령을 실행하려면 다음을 수행해야 합니다.

- AWS CLI를 설치하고 구성합니다. 자세한 내용은 설치 AWS CLI 및 에 대한 인증 및 액세스 자격 증명 AWS CLI 단원을 참조하세요.
- Amazon EC2 액세스를 허용하도록 IAM 권한을 설정합니다. Amazon 에 대한 IAM 권한에 대한 자세한 내용은 Amazon EC2 사용 설명서의 Amazon 정책을 EC2참조하세요. IAM EC2
- 키 페어 및 보안 그룹을 생성합니다.
- Amazon Machine Image(AMI)를 선택하고 AMI ID를 기록합니다. 자세한 내용은 Amazon EC2 사용 설명서의 적합한 찾기를 AMI 참조하세요.

## 인스턴스 시작

AMI 선택한 를 사용하여 Amazon EC2 인스턴스를 시작하려면 `aws ec2 run-instances` 명령을 사용합니다. 인스턴스를 가상 프라이빗 클라우드()로 시작할 수 있습니다VPC.

처음에는 인스턴스가 `pending` 상태로 표시되지만 몇 분 후에 `running` 상태로 변경됩니다.

다음 예제에서는 의 지정된 서브넷에서 `t2.micro` 인스턴스를 시작하는 방법을 보여줍니다VPC. 교체 *`italicized`* 파라미터 값은 자체 파라미터 값입니다.

```
$ aws ec2 run-instances --image-id ami-xxxxxxxx --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
    }
  ]
}
```

인스턴스에 블록 디바이스 추가

실행된 각 인스턴스에는 연관된 루트 디바이스 볼륨이 있습니다. 블록 디바이스 매핑을 사용하여 추가 Amazon Elastic Block Store(AmazonEBS) 볼륨 또는 인스턴스 스토어 볼륨을 지정하여 인스턴스가 시작될 때 인스턴스에 연결할 수 있습니다.

인스턴스에 블록 디바이스를 추가하려면 `run-instances`를 사용할 때 `--block-device-mappings` 옵션을 지정합니다.

다음 예제 파라미터는 크기가 20GB인 표준 Amazon EBS 볼륨을 프로비저닝하고 식별자 를 사용하여 인스턴스에 매핑합니다`/dev/sdf`.

```
--block-device-mappings "[{\"DeviceName\":\"/dev/sdf\",\"Ebs\":{\"VolumeSize\":20,
\"DeleteOnTermination\":false}}]"
```

다음 예제에서는 기존 스냅샷을 `/dev/sdf`기반으로 에 매핑된 Amazon EBS 볼륨을 추가합니다. 스냅샷은 볼륨에 로드되는 이미지를 나타냅니다. 스냅샷을 지정할 때 볼륨 크기를 지정할 필요가 없습니다. 이미지를 담을 만큼 충분히 큽니다. 그러나 크기를 지정하는 경우 스냅샷의 크기보다 크거나 같아야 합니다.

```
--block-device-mappings "[{\"DeviceName\":\"/dev/sdf\",\"Ebs\":{\"SnapshotId\":\"snap-a1b2c3d4\"}}]"
```

다음 예제에서는 인스턴스에 두 개의 볼륨을 추가합니다. 인스턴스에 사용 가능한 볼륨 수는 인스턴스 유형에 따라 다릅니다.

```
--block-device-mappings "[{\"DeviceName\":\"/dev/sdf\",\"VirtualName\":\"ephemeral0\"},{\"DeviceName\":\"/dev/sdg\",\"VirtualName\":\"ephemeral1\"}]"
```

다음 예제에서는 매핑(/dev/sdj)을 생성하지만 인스턴스에 볼륨을 프로비저닝하지 않습니다.

```
--block-device-mappings "[{\"DeviceName\":\"/dev/sdj\",\"NoDevice\":\"\"}]"
```

자세한 내용은 Amazon EC2 사용 설명서의 디바이스 매핑 차단을 참조하세요.

인스턴스에 태그 추가

태그는 AWS 리소스에 할당하는 레이블입니다. 이를 통해 다양한 용도로 사용할 수 있는 메타데이터를 리소스에 추가할 수 있습니다. 자세한 내용은 Amazon EC2 사용 설명서의 리소스 태그 지정을 참조하세요.

다음 예제에서는 `aws ec2 create-tags` 명령을 사용하여 키 이름이 "Name"이고 값이 "MyInstance"인 태그를 지정된 인스턴스에 추가하는 방법을 보여줍니다.

```
$ aws ec2 create-tags --resources i-5203422c --tags Key=Name,Value=MyInstance
```

인스턴스에 연결합니다

인스턴스가 실행될 때 실행 중인 인스턴스에 연결하여 바로 앞에 있는 컴퓨터를 사용하는 것처럼 인스턴스를 사용할 수 있습니다. 자세한 내용은 Amazon 사용 설명서의 Amazon EC2 인스턴스에 연결을 참조하세요. EC2

인스턴스 나열

를 사용하여 인스턴스 AWS CLI 를 나열하고 인스턴스에 대한 정보를 볼 수 있습니다. 모든 인스턴스를 나열하거나 관심이 있는 인스턴스에 따라 결과를 필터링할 수 있습니다.

다음 예제에서는 `aws ec2 describe-instances` 명령을 사용하는 방법을 보여줍니다.

다음 명령은 모든 인스턴스를 나열합니다.

```
$ aws ec2 describe-instances
```

다음 명령은 목록을 t2.micro 인스턴스로만 필터링하고 각 매치에 대한 InstanceId 값만 출력합니다.

```
$ aws ec2 describe-instances --filters "Name=instance-type,Values=t2.micro" --query
 "Reservations[].Instances[].InstanceId"
[
    "i-05e998023d9c69f9a"
]
```

다음 명령은 Name=MyInstance 태그가 있는 인스턴스를 나열합니다.

```
$ aws ec2 describe-instances --filters "Name=tag:Name,Values=MyInstance"
```

다음 명령은 ami-x0123456, AMIs및 중 하나를 사용하여 시작된 인스턴스를 나열합니다ami-y0123456ami-z0123456.

```
$ aws ec2 describe-instances --filters "Name=image-id,Values=ami-x0123456,ami-
y0123456,ami-z0123456"
```

인스턴스 종료

인스턴스를 종료하면 삭제됩니다. 인스턴스를 종료하면 인스턴스에 다시 연결할 수 없습니다.

인스턴스 상태가 shutting-down 또는 terminated로 변경되는 즉시 해당 인스턴스에 대한 반복적인 요금 부과가 중단됩니다. 나중에 인스턴스에 다시 연결하려면 terminate-instances 대신 stop-instances를 사용합니다. 자세한 내용은 Amazon EC2 사용 설명서의 인스턴스 종료를 참조하세요.

인스턴스를 삭제하려면 aws ec2 terminate-instances 명령을 사용하여 삭제합니다.

```
$ aws ec2 terminate-instances --instance-ids i-5203422c
{
    "TerminatingInstances": [
        {
            "InstanceId": "i-5203422c",
            "CurrentState": {
                "Code": 32,
                "Name": "shutting-down"
            },
```

```
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
```

참조

AWS CLI 참조:

- aws ec2
- aws ec2 create-tags
- aws ec2 describe-instances
- aws ec2 run-instances
- aws ec2 terminate-instances

기타 참조:

- Amazon Elastic Compute Cloud 설명서
- 및 AWS CLI 코드 예제를 AWS SDK 보고 기여하려면 의 AWS 코드 예제 리포지토리를 참조하세요GitHub.

## 에서 bash 스크립트로 Amazon EC2 인스턴스 유형 변경 AWS CLI

Amazon에 대한 이 bash 스크립팅 예제는 AWS Command Line Interface ()를 사용하여 Amazon 인스턴스의 EC2 인스턴스 유형을 EC2 변경합니다AWS CLI. 인스턴스가 실행 중인 경우 인스턴스를 중지하고, 인스턴스 유형을 변경한 다음, 요청된 경우 인스턴스를 다시 시작합니다. 셸 스크립트는 명령줄 인터페이스에서 실행되도록 설계된 프로그램입니다.

> **Note**
>
> 추가 명령 예제는 AWS CLI 참조 가이드 를 참조하세요.

주제

- 시작하기 전에

- 이 예제 정보
- 파라미터
- 파일
- 참조

시작하기 전에

아래 예제 중 하나를 실행하려면 먼저 다음 작업을 완료해야 합니다.

- AWS CLI를 설치하고 구성합니다. 자세한 내용은 설치 AWS CLI 및 에 대한 인증 및 액세스 자격 증명 AWS CLI 단원을 참조하세요.
- 사용하는 프로필에는 예제에서 수행되는 AWS 작업을 허용하는 권한이 있어야 합니다.
- 중지 및 수정할 권한이 있는 계정에서 실행 중인 Amazon EC2 인스턴스입니다. 테스트 스크립트를 실행하면 테스트 스크립트가 인스턴스를 시작하고 유형을 변경하여 인스턴스를 테스트한 다음 인스턴스를 종료합니다.
- AWS 가장 좋은 방법은 이 코드에 최소 권한 또는 작업을 수행하는 데 필요한 권한만 부여하는 것입니다. 자세한 내용은 AWS 자격 증명 및 액세스 관리(IAM) 사용 설명서의 최소 권한 부여를 참조하세요.
- 이 코드는 일부 AWS 리전에서 테스트되지 않았습니다. 일부 AWS 서비스는 특정 리전에서만 사용할 수 있습니다. 자세한 내용은 AWS 일반 참조 안내서에서 서비스 엔드포인트 및 할당량을 참조하세요.
- 이 코드를 실행하면 AWS 계정에 요금이 부과될 수 있습니다. 이 스크립트에 의해 생성된 모든 리소스를 사용한 후 제거하는 것은 사용자의 책임입니다.

이 예제 정보

이 예제는 다른 스크립트나 명령줄에서 `source`할 수 있는 셸 스크립트 파일 `change_ec2_instance_type.sh`의 함수로 작성됩니다. 각 스크립트 파일에는 각 함수를 설명하는 주석이 들어 있습니다. 함수가 메모리에 있으면 명령줄에서 함수를 호출할 수 있습니다. 예를 들어, 다음 명령은 지정된 인스턴스의 유형을 `t2.nano`로 변경합니다.

```
$ source ./change_ec2_instance_type.sh
$ ./change_ec2_instance_type -i *instance-id* -t new-type
```

전체 예제 및 다운로드 가능한 스크립트 파일은 의 AWS 코드 예제 리포지토리에서 Amazon EC2 인스턴스 유형 변경을 참조하세요GitHub.

파라미터

-i - (문자열) 수정할 인스턴스 ID를 지정합니다.

-t - (문자열) 전환할 Amazon EC2 인스턴스 유형을 지정합니다.

-r - (스위치) 기본적으로 설정되지 않습니다. `-r`이 설정된 경우 유형 스위치 뒤에 인스턴스를 다시 시작합니다.

-f - (스위치) 기본적으로 스크립트는 스위치를 만들기 전에 인스턴스를 종료할지 확인하는 메시지를 사용자에게 표시합니다. `-f`가 설정된 경우, 함수는 유형 스위치를 만들기 위해 인스턴스를 종료하기 전에 사용자에게 메시지를 표시하지 않습니다

-v - (스위치) 기본적으로 스크립트는 자동으로 작동하며 오류가 발생한 경우에만 출력을 표시합니다. `-v`가 설정된 경우 함수는 작업 전체 상태를 표시합니다.

파일

**`change_ec2_instance_type.sh`**

기본 스크립트 파일에는 다음 작업을 수행하는 `change_ec2_instance_type()` 함수가 포함되어 있습니다.

- 지정된 Amazon EC2 인스턴스가 존재하는지 확인합니다.
- `-f`를 선택하지 않으면 인스턴스를 중지하기 전에 사용자에게 경고합니다.
- 인스턴스 유형을 변경합니다.
- `-r`을 설정하면 인스턴스를 다시 시작하고 인스턴스가 실행 중인지 확인합니다.

에서 [change_ec2_instance_type.sh](#) 에 대한 코드를 확인합니다GitHub.

**`test_change_ec2_instance_type.sh`**

파일 `test_change_ec2_instance_type.sh` 스크립트는 `change_ec2_instance_type` 함수에 대한 다양한 코드 경로를 테스트합니다. 테스트 스크립트의 모든 단계가 올바르게 작동하는 경우 테스트 스크립트는 생성한 모든 리소스를 제거합니다.

다음 파라미터와 함께 테스트 스크립트를 실행할 수 있습니다.

- -v - (스위치) 각 테스트는 pass/failure status as they run. By default, the tests runs silently and the output includes only the final overall pass/failure 상태를 표시합니다.

- -i - (스위치) 각 테스트 후에 스크립트가 일시 중지되어 각 단계의 중간 결과를 찾아볼 수 있습니다. Amazon EC2 콘솔을 사용하여 인스턴스의 현재 상태를 검사할 수 있습니다. 프롬프트ENTER에서 를 누르면 스크립트가 다음 단계로 진행됩니다.

에서 test_change_ec2_instance_type.sh 에 대한 코드를 확인합니다GitHub.

**awsdocs_general.sh**

스크립트 파일 awsdocs_general.sh에는 AWS CLI에 대한 고급 예제에서 사용되는 범용 함수가 들어 있습니다.

에서 awsdocs_general.sh 에 대한 코드를 확인합니다GitHub.

참조

AWS CLI 참조:

- 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

기타 참조:

- Amazon Elastic Compute Cloud 설명서
- 및 AWS CLI 코드 예제를 AWS SDK 보고 기여하려면 의 AWS 코드 예제 리포지토리를 참조하세요GitHub.

## 에서 Amazon S3 Glacier 사용 AWS CLI

| Amazon S3 Glacier 소개 |
| --- |
| Amazon S3 Glacier 소개 |

이 주제에서는 S3 Glacier에 대한 일반적인 작업을 수행하는 AWS CLI 명령의 예를 보여줍니다. 이 예제에서는 를 사용하여 AWS CLI 대용량 파일을 작은 부분으로 분할하고 명령줄에서 업로드하여 S3 Glacier에 업로드하는 방법을 보여줍니다.

AWS Command Line Interface ()를 사용하여 Amazon S3 Glacier 기능에 액세스할 수 있습니다AWS CLI. S3 Glacier에 대한 AWS CLI 명령을 나열하려면 다음 명령을 사용합니다.

```
aws glacier help
```

> **Note**
>
> 명령 참조 및 추가 예제는 AWS CLI 명령 참조의 `aws glacier` 단원을 참조하세요.

주제



## 사전 조건

`glacier` 명령을 실행하려면 다음을 수행해야 합니다.

- AWS CLI를 설치하고 구성합니다. 자세한 내용은 설치 AWS CLI 및 에 대한 인증 및 액세스 자격 증명 AWS CLI 단원을 참조하세요.
- 이 자습서에서는 Linux 및 macOS를 비롯한 UNIX 계열 운영 체제에 일반적으로 사전 설치된 몇 가지 명령줄 도구를 사용합니다. Windows 사용자는 Cygwin을 설치하고 Cygwin 터미널에서 명령을 실행하여 동일한 도구를 사용할 수 있습니다. 동일한 기능을 수행하는 Windows 기본 명령 및 유틸리티가 표시되어 있습니다(사용 가능한 경우).

## Amazon S3 Glacier 볼트 생성

`create-vault` 명령을 사용하여 볼트를 생성합니다.

```
$ aws glacier create-vault --account-id - --vault-name myvault
{
    "location": "/123456789012/vaults/myvault"
}
```

> **Note**
>
> 모든 S3 Glacier 명령에는 계정 ID 파라미터가 필요합니다. 현재 계정을 사용하려면 하이픈 문자(`--account-id -`)를 사용합니다.

## 파일 업로드 준비

테스트 업로드를 위한 파일을 생성합니다. 다음 명령은 라는 파일을 생성합니다.*largefile* 에는 정확히 3MiB의 무작위 데이터가 포함되어 있습니다.

Linux 또는 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
```

dd는 입력 파일에서 출력 파일로 많은 바이트를 복사하는 유틸리티입니다. 앞의 예제에서는 시스템 디바이스 파일 /dev/urandom을 임의 데이터 소스로 사용합니다. `fsutil`은 Windows에서 유사한 함수를 수행합니다.

Windows

```
C:\> fsutil file createnew largefile 3145728
File C:\temp\largefile is created
```

그런 다음 파일 분할기를 사용하여 파일을 1MiB(1,048,576바이트) 청크로 분할합니다.

```
$ split -b 1048576 --verbose largefile chunk
creating file `chunkaa'
creating file `chunkab'
creating file `chunkac'
```

## 멀티파트 업로드 및 파일 업로드 시작

initiate-multipart-upload 명령을 사용하여 Amazon S3 Glacier에서 멀티파트 업로드를 생성합니다.

```
$ aws glacier initiate-multipart-upload --account-id - --archive-description "multipart
 upload test" --part-size 1048576 --vault-name myvault
{
    "uploadId": "19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-
OssZtLqyFu7sY1_lR7vgFuJV6NtcV5zpsJ",
    "location": "/123456789012/vaults/myvault/multipart-
uploads/19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-
OssZtLqyFu7sY1_lR7vgFuJV6NtcV5zpsJ"
}
```

S3 Glacier에서 멀티파트 업로드를 구성하려면 각 파트의 크기(바이트, 이 예제에서는 1MiB), 볼트 이름 및 계정 ID가 필요합니다. 는 작업이 완료되면 업로드 ID를 AWS CLI 출력합니다. 나중에 사용하기 위해 업로드 ID를 셸 변수에 저장합니다.

Linux 또는 macOS

```
$ UPLOADID="19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-
OssZtLqyFu7sY1_lR7vgFuJV6NtcV5zpsJ"
```

Windows

```
C:\> set UPLOADID="19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-
OssZtLqyFu7sY1_lR7vgFuJV6NtcV5zpsJ"
```

그런 다음 upload-multipart-part 명령을 사용하여 세 개의 파트를 각각 업로드합니다.

```
$ 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**
>
> 앞의 예에서는 Linux에서 달러 기호($)를 사용하여 UPLOADID 셸 변수의 내용을 참조합니다. Windows 명령줄에서는 변수 이름의 양쪽에 퍼센트 기호(%)를 사용합니다(예: %UPLOADID%).

S3 Glacier에서 올바른 순서로 다시 수집할 수 있도록 각 파트를 업로드할 때 각 파트의 바이트 범위를 지정해야 합니다. 각 조각은 1,048,576바이트입니다. 따라서 첫 번째 조각은 0-1048575바이트, 두 번째 조각은 1048576-2097151바이트, 세 번째 조각은 2097152-3145727바이트를 차지합니다.

## 업로드 완료

Amazon S3 Glacier는 업로드된 모든 조각이 AWS 정상에 도달했는지 확인하기 위해 원본 파일의 트리 해시가 필요합니다.

트리 해시를 계산하려면 파일을 1MiB 부분으로 분할하고 각 조각의 바이너리 SHA-256해시를 계산해야 합니다. 그런 다음 해시 목록을 쌍으로 분할하고, 2개의 이진 해시를 각 쌍으로 결합하며, 결과의 해시를 가져옵니다. 하나의 해시만 남을 때까지 이 프로세스를 반복합니다. 임의 레벨에서 홀수 해시가 있을 경우 수정하지 않고 다음 레벨로 승격시킵니다.

명령줄 유틸리티를 사용할 때 트리 해시를 올바로 계산하는 핵심은 각 해시를 이진 형식으로 저장하고 마지막 단계에서만 16진수로 변환하는 것입니다. 트리의 16진수 버전 해시를 결합하거나 해시할 경우 잘못된 결과가 발생할 수 있습니다.

> **Note**
>
> Windows 사용자는 `type` 대신 `cat` 명령을 사용할 수 있습니다. OpenSSL은 Open SSL.org에서 Windows에 사용할 수 있습니다.

트리 해시를 계산하려면

1. 아직 분할하지 않은 경우, 원본 파일을 1MiB로 분할합니다.

```
$ split --bytes=1048576 --verbose largefile chunk
creating file `chunkaa'
creating file `chunkab'
creating file `chunkac'
```

2. 각 청크의 바이너리 SHA-256 해시를 계산하고 저장합니다.

```
$ openssl dgst -sha256 -binary chunkaa > hash1
$ openssl dgst -sha256 -binary chunkab > hash2
$ openssl dgst -sha256 -binary chunkac > hash3
```

3. 처음 2개 해시를 결합하고 결과의 이진 해시를 가져옵니다.

```
$ cat hash1 hash2 > hash12
$ openssl dgst -sha256 -binary hash12 > hash12hash
```

4. 청크 aa 및 ab의 상위 해시를 청크 ac의 해시와 결합하고 결과를 해시합니다. 이때는 16진수가 출력됩니다. 결과를 셸 변수에 저장합니다.

```
$ cat hash12hash hash3 > hash123
$ openssl dgst -sha256 hash123
SHA256(hash123)= 9628195fcdbcbbe76cdde932d4646fa7de5f219fb39823836d81f0cc0e18aa67
$ TREEHASH=9628195fcdbcbbe76cdde932d4646fa7de5f219fb39823836d81f0cc0e18aa67
```

마지막으로 complete-multipart-upload 명령을 사용하여 업로드를 완료합니다. 이 명령에서는 원본 파일의 크기(바이트), 최종 트리 해시 값(16진수) 및 계정 ID와 볼트 이름을 사용합니다.

```
$ aws glacier complete-multipart-upload --checksum $TREEHASH --archive-size 3145728 --
upload-id $UPLOADID --account-id - --vault-name myvault
{
    "archiveId": "d3AbWhE0YE1m6f_fI1jPG82F8xzbMEEZmrAlLGAAONJAzo5QdP-
N83MKqd96Unspoa5H5lItWX-sK8-QS0ZhwsyGiu9-R-
kwWUyS1dSBlmgPPWkEbeFfqDSav053rU7FvVLHfRc6hg",
    "checksum": "9628195fcdbcbbe76cdde932d4646fa7de5f219fb39823836d81f0cc0e18aa67",
    "location": "/123456789012/vaults/myvault/archives/
d3AbWhE0YE1m6f_fI1jPG82F8xzbMEEZmrAlLGAAONJAzo5QdP-N83MKqd96Unspoa5H5lItWX-sK8-
QS0ZhwsyGiu9-R-kwWUyS1dSBlmgPPWkEbeFfqDSav053rU7FvVLHfRc6hg"
}
```

describe-vault 명령을 사용하여 볼트 상태를 확인할 수도 있습니다.

```
$ 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**
>
> 볼트 상태는 매일 한 번 정도 업데이트됩니다. 자세한 내용은 볼트 작업을 참조하세요.

이제 생성한 청크 및 해시 파일을 안전하게 제거할 수 있습니다.

```
$ rm chunk* hash*
```

멀티파트 업로드에 대한 자세한 내용은 Amazon S3 Glacier 개발자 안내서에서 파트로 대용량 아카이브 업로드 및 체크섬 컴퓨팅을 참조하세요.

## 리소스

AWS CLI 참조:

- aws glacier
- aws glacier complete-multipart-upload
- aws glacier create-vault
- aws glacier describe-vault
- aws glacier initiate-multipart-upload

서비스 참조:

- Amazon S3 Glacier 개발자 안내서
- Amazon S3 Glacier 개발자 안내서의 대용량 아카이브를 여러 부분으로 나누어 업로드
- Amazon S3 Glacier 개발자 안내서의 체크섬 계산
- Amazon S3 Glacier 개발자 안내서의 볼트 작업

# IAM 에서 사용 AWS CLI

에 대한 소개 AWS Identity and Access Management

에 대한 소개 AWS Identity and Access Management

AWS Identity and Access Management (IAM)를 사용하여 AWS Command Line Interface ()의 기능에 액세스할 수 있습니다AWS CLI. 에 대한 AWS CLI 명령을 나열하려면 다음 명령을 IAM사용합니다.

```
aws iam help
```

이 주제에서는 에 대한 일반적인 작업을 수행하는 AWS CLI 명령의 예를 보여줍니다IAM.

명령을 실행하기 전에 기본 자격 증명을 설정합니다. 자세한 내용은 에 대한 설정 구성 AWS CLI 단원을 참조하십시오.

IAM 서비스에 대한 자세한 내용은 AWS Identity and Access Management 사용 설명서 섹션을 참조하세요.

주제

- IAM 사용자 및 그룹 생성
- 사용자에게 IAM 관리형 정책 연결
- IAM 사용자의 초기 암호 설정
- IAM 사용자의 액세스 키 생성

## IAM 사용자 및 그룹 생성

그룹을 생성하고 이 그룹에 새 사용자를 추가하려면

1. `create-group` 명령을 사용하여 그룹을 생성합니다.

```
$ 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. create-user 명령을 사용하여 사용자를 생성합니다.

```
$ 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. add-user-to-group 명령을 사용하여 사용자를 그룹에 추가합니다.

```
$ aws iam add-user-to-group --user-name MyUser --group-name MyIamGroup
```

4. MyIamGroup 그룹에 MyUser가 포함되어 있는지 확인하려면 get-group 명령을 사용합니다.

```
$ 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"
}
```

## 사용자에게 IAM 관리형 정책 연결

이 예제의 정책은 사용자에게 "파워 유저 액세스"를 제공합니다.

사용자에게 IAM 관리형 정책을 연결하려면

1. 연결할 정책의 Amazon 리소스 이름(ARN)을 결정합니다. 다음 명령은 `list-policies`를 사용하여 이름이 인 ARN 정책의 를 찾습니다`PowerUserAccess`. 그런 다음 환경 변수ARN에 저장합니다.

```
$ export POLICYARN=$(aws iam list-policies --query 'Policies[?
PolicyName==`PowerUserAccess`].{ARN:Arn}' --output text)       ~
$ echo $POLICYARN
arn:aws:iam::aws:policy/PowerUserAccess
```

2. 정책을 연결하려면 attach-user-policyattach-user-policy 명령을 사용하고 정책 를 포함하는 환경 변수를 참조합니다ARN.

```
$ aws iam attach-user-policy --user-name MyUser --policy-arn $POLICYARN
```

3. list-attached-user-policies 명령을 실행하여 정책이 사용자에게 연결되었는지 확인합니다.

```
$ aws iam list-attached-user-policies --user-name MyUser
{
    "AttachedPolicies": [
        {
            "PolicyName": "PowerUserAccess",
            "PolicyArn": "arn:aws:iam::aws:policy/PowerUserAccess"
        }
    ]
}
```

자세한 내용은 액세스 관리 리소스를 참조하세요. 이 주제에서는 권한 및 정책 개요에 대한 링크와 Amazon S3, Amazon EC2및 기타 서비스에 액세스하기 위한 정책 예제에 대한 링크를 제공합니다.

## IAM 사용자의 초기 암호 설정

다음 명령은 create-login-profile을 사용하여 지정된 사용자의 초기 암호를 설정합니다. 처음으로 로그인한 사용자는 본인만 아는 암호로 변경해야 합니다.

```
$ 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
    }
}
```

update-login-profile 명령을 사용하여 사용자의 암호를 변경합니다.

```
$ aws iam update-login-profile --user-name MyUser --password My!User1ADifferentP@ssword
```

## IAM 사용자의 액세스 키 생성

create-access-key 명령을 사용하여 사용자를 위한 액세스 키를 생성할 수 있습니다. 액세스 키는 액세스 키 ID와 비밀 키로 구성된 보안 자격 증명 세트입니다.

사용자는 한 번에 두 개의 액세스 키만 생성할 수 있습니다. 세 번째 세트를 생성하려 할 경우 이 명령은 LimitExceeded 오류를 반환합니다.

```
$ 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"
    }
}
```

delete-access-key 명령을 사용하여 사용자를 위한 액세스 키를 삭제합니다. 액세스 키 ID를 사용하여 삭제할 액세스 키를 지정합니다.

```
$ aws iam delete-access-key --user-name MyUser --access-key-id AKIAIOSFODNN7EXAMPLE
```

# 에서 Amazon S3 사용 AWS CLI

Amazon Simple Storage Service(Amazon S3) 소개

Amazon Simple Storage Service(Amazon S3) 소개 - AWS의 클라우드 스토리지

AWS Command Line Interface ()를 사용하여 Amazon Simple Storage Service(Amazon S3)의 기능에 액세스할 수 있습니다AWS CLI. Amazon S3는 확장성과 내구성이 뛰어난 객체 스토리지 서비스입니다. Amazon S3는 거의 무제한의 스토리지 용량을 제공하도록 설계되었으므로 다양한 데이터 스토리지 및 관리 요구 사항에 이상적인 솔루션입니다.

Amazon S3를 사용하면 작은 파일에서 큰 데이터 세트에 이르기까지 모든 양의 데이터를 객체 형태로 저장하고 검색할 수 있습니다. 각 객체는 버킷이라는 컨테이너에 저장되며, 버킷은 를 통해 액세스하고 관리 AWS Management Console 하거나 SDKs, 도구 및 를 AWS 통해 프로그래밍 방식으로 관리할 수 있습니다 AWS CLI.

기본 스토리지를 포함하여 Amazon S3는 수명 주기 관리, 버전 관리, 확장성 및 보안을 비롯한 다양한 기능도 제공합니다. 이러한 솔루션은 다른 AWS 서비스 솔루션과 통합되어 필요에 맞게 확장 가능한 클라우드 기반 솔루션을 구축할 수 있습니다.

는 Amazon S3에 액세스하기 위한 두 계층의 명령을 AWS CLI 제공합니다.

- s3 - 객체 및 버킷 생성, 조작, 삭제 및 동기화와 같은 일반적인 작업 수행을 단순화 AWS CLI 하는 를 위해 특별히 만들어진 사용자 지정 상위 수준 명령입니다.
- s3api – 고급 API 작업을 수행할 수 있는 모든 Amazon S3 작업에 대한 직접 액세스를 제공합니다.

이 설명서의 주제는 다음과 같습니다.

- 에서 상위 수준 (s3) 명령 사용 AWS CLI
- 에서 API-Level(s3api) 명령 사용 AWS CLI
- 의 Amazon S3 버킷 수명 주기에 대한 스크립팅 예제 AWS CLI

## 에서 상위 수준 (s3) 명령 사용 AWS CLI

이 주제에서는 AWS CLI에서 `aws s3` 명령을 사용하여 Amazon S3 버킷과 객체를 관리하는 데 사용할 수 있는 몇 가지 명령을 설명합니다. 이 주제에서 다루지 않은 명령과 추가 명령 예제는 AWS CLI 참조에 있는 `aws s3` 명령을 참조하세요.

상위 수준 aws s3 명령은 Amazon S3 객체 관리를 간소화합니다. 이 명령을 사용하면 명령 자체 내에서와 로컬 디렉터리를 사용하여 Amazon S3의 내용을 관리할 수 있습니다.

주제



## 사전 조건

s3 명령을 실행하려면 다음을 수행해야 합니다.

- AWS CLI를 설치하고 구성합니다. 자세한 내용은 설치 AWS CLI 및 에 대한 인증 및 액세스 자격 증명 AWS CLI 단원을 참조하세요.
- 사용하는 프로필에는 예제에서 수행하는 AWS 작업을 허용하는 권한이 있어야 합니다.
- 다음 Amazon S3 용어를 이해하세요.
  - 버킷 - 최상위 Amazon S3 폴더입니다.
  - 접두사 - 버킷의 Amazon S3 폴더입니다.
  - 객체 - Amazon S3 버킷에서 호스팅되는 모든 항목입니다.

## 시작하기 전에

이 섹션에서는 aws s3 명령을 사용하기 전에 주의해야 할 몇 가지 사항에 대해 설명합니다.

대용량 객체 업로드

aws s3 명령을 사용하여 Amazon S3 버킷에 대용량 객체를 업로드하면 AWS CLI 에서 자동으로 멀티파트 업로드를 수행합니다. 이러한 aws s3 명령을 사용할 때는 실패한 업로드를 재개할 수 없습니다.

제한 시간으로 인해 멀티파트 업로드가 실패하거나 에서 수동으로 취소한 경우 AWS CLI는 업로드를 AWS CLI 중지하고 생성된 파일을 정리합니다. 이 프로세스는 몇 분 정도 걸릴 수 있습니다.

kill 명령이나 시스템 오류로 인해 멀티파트 업로드 또는 정리 프로세스가 취소되면 생성된 파일은 Amazon S3 버킷에 남아 있습니다. 멀티파트 업로드를 정리하려면 s3api abort-multipart-upload 명령을 사용합니다.

멀티파트 복사의 파일 속성 및 태그

aws s3 네임스페이스의 AWS CLI 버전 1 버전의 명령을 사용하여 한 Amazon S3 버킷 위치에서 다른 Amazon S3 버킷 위치로 파일을 복사하고 해당 작업에서 멀티파트 복사 를 사용하는 경우 소스 객체의 파일 속성은 대상 객체로 복사되지 않습니다.

버킷 만들기

s3 mb 명령을 사용하여 버킷을 만듭니다. 버킷 이름은 전역적으로 고유해야 하며(모든 Amazon S3에서 고유) DNS 규정을 준수해야 합니다.

버킷 이름에는 소문자, 숫자, 하이픈, 마침표가 포함될 수 있습니다. 버킷 이름은 문자나 숫자로만 시작하고 끝날 수 있으며 하이픈이나 다른 마침표 옆에 마침표가 포함될 수 없습니다.

구문

```
$ aws s3 mb <target> [--options]
```

s3 mb 예제

다음 예제에서는 s3://amzn-s3-demo-bucket 버킷을 생성합니다.

```
$ aws s3 mb s3://amzn-s3-demo-bucket
```

버킷 및 객체 나열

버킷, 폴더 또는 객체를 나열하려면 s3 ls 명령을 사용합니다. 대상 또는 옵션 없이 명령을 사용하면 모든 버킷이 나열됩니다.

구문

```
$ aws s3 ls <target> [--options]
```

이 명령과 함께 사용할 몇 가지 일반적인 옵션 및 예제는 s3 명령에 자주 사용되는 옵션 섹션을 참조하세요. 사용 가능한 옵션의 전체 목록은 AWS CLI 명령 참조에서 `s3 ls` 단원을 참조하세요.

s3 ls 예제

다음 예제에서는 모든 Amazon S3 버킷을 나열합니다.

```
$ aws s3 ls
2018-12-11 17:08:50 amzn-s3-demo-bucketamzn-s3-demo-bucket1
2018-12-14 14:55:44 amzn-s3-demo-bucket2
```

다음 명령은 버킷에 있는 모든 객체와 접두사를 나열합니다. 이 예제 출력에서 접두사 `example/`에는 `MyFile1.txt`라는 이름의 파일이 하나 있습니다.

```
$ aws s3 ls s3://amzn-s3-demo-bucket
                           PRE example/
2018-12-04 19:05:48          3 MyFile1.txt
```

명령에 특정 접두사를 포함하여 출력을 필터링할 수 있습니다. 다음 명령은 의 객체를 나열합니다.*bucket-name/example/* (즉, 의 객체 *bucket-name* 접두사로 필터링 *example/*).

```
$ aws s3 ls s3://amzn-s3-demo-bucket/example/
2018-12-06 18:59:32          3 MyFile1.txt
```

버킷 삭제

버킷을 삭제하려면 `s3 rb` 명령을 사용합니다.

구문

```
$ aws s3 rb <target> [--options]
```

s3 rb 예제

다음 예제에서는 `s3://amzn-s3-demo-bucket` 버킷을 제거합니다.

```
$ aws s3 rb s3://amzn-s3-demo-bucket
```

기본적으로 작업에 성공하려면 버킷이 비어 있어야 합니다. 비어 있지 않은 버킷을 제거하려면 `--force` 옵션을 포함시켜야 합니다. 이전에 삭제했지만 보관된 객체가 포함되어 있는 버전 지정된 버킷을 사용할 경우 이 명령을 사용하여 버킷을 제거할 수 없습니다. 먼저 모든 내용을 제거해야 합니다.

다음 예제에서는 버킷의 모든 객체와 접두사를 삭제한 다음 버킷을 삭제합니다.

```
$ aws s3 rb s3://amzn-s3-demo-bucket --force
```

객체 삭제

버킷이나 로컬 디렉터리의 객체를 삭제하려면 `s3 rm` 명령을 사용합니다.

구문

```
$ aws s3 rm  <target> [--options]
```

이 명령과 함께 사용할 몇 가지 일반적인 옵션 및 예제는 s3 명령에 자주 사용되는 옵션 섹션을 참조하세요. 전체 옵션 목록은 AWS CLI 명령 참조에서 `s3 rm` 단원을 참조하세요.

s3 rm 예제

다음 예제에서는 `filename.txt`에서 `s3://amzn-s3-demo-bucket/example` 파일을 삭제합니다.

```
$ aws s3 rm s3://amzn-s3-demo-bucket/example/filename.txt
```

다음 예제에서는 `s3://amzn-s3-demo-bucket/example` 옵션을 사용하여 `--recursive`에서 모든 객체를 삭제합니다.

```
$ aws s3 rm s3://amzn-s3-demo-bucket/example --recursive
```

객체 이동

`s3 mv` 명령을 사용하여 버킷이나 로컬 디렉터리에서 객체를 이동합니다. `s3 mv` 명령은 소스 객체 또는 파일을 지정된 대상으로 복사한 다음 소스 객체 또는 파일을 삭제합니다.

구문

```
$ aws s3 mv <source> <target> [--options]
```

이 명령과 함께 사용할 몇 가지 일반적인 옵션 및 예제는 s3 명령에 자주 사용되는 옵션 섹션을 참조하세요. 사용 가능한 옵션의 전체 목록은 AWS CLI 명령 참조에서 `s3 mv` 단원을 참조하세요.

> **Warning**
>
> Amazon S3 소스 ARNs 또는 대상 에서 모든 유형의 액세스 포인트 또는 액세스 포인트 별칭을 사용하는 경우 소스 및 대상 Amazon S3가 다른 기본 버킷으로 URIs 확인되도록 각별히 주의해야 URIs합니다. 소스 버킷과 대상 버킷이 동일한 경우 소스 파일 또는 객체를 자체로 이동할 수 있으므로 소스 파일 또는 객체가 실수로 삭제될 수 있습니다. 소스 버킷과 대상 버킷이 동일하지 않은지 확인하려면 `--validate-same-s3-paths` 파라미터를 사용하거나 환경 변수를 `AWS_CLI_S3_MV_VALIDATE_SAME_S3_PATHS`로 설정합니다`true`.

s3 mv 예제

다음 예제에서는 `s3://amzn-s3-demo-bucket/example`에서 모든 객체를 `s3://amzn-s3-demo-bucket/`으로 이동합니다.

```
$ aws s3 mv s3://amzn-s3-demo-bucket/example s3://amzn-s3-demo-bucket/
```

다음 예제에서는 `s3 mv` 명령을 사용하여 현재 작업 디렉터리에서 Amazon S3 버킷으로 로컬 파일을 이동합니다.

```
$ aws s3 mv filename.txt s3://amzn-s3-demo-bucket
```

다음 예제에서는 Amazon S3 버킷에서 현재 작업 디렉터리로 파일을 이동합니다. 여기서 `./`는 현재 작업 디렉터리를 지정합니다.

```
$ aws s3 mv s3://amzn-s3-demo-bucket/filename.txt ./
```

객체 복사

`s3 cp` 명령을 사용하여 버킷이나 로컬 디렉터리에서 객체를 복사합니다.

구문

```
$ aws s3 cp <source> <target> [--options]
```

표준 입력(stdin) 또는 표준 출력(stdout)으로의 파일 스트리밍을 위해 dash 파라미터를 사용할 수 있습니다.

> **Warning**
>
> 를 사용하는 경우 PowerShell쉘이 의 인코딩을 변경하거나 파이프 입력 CRLF 또는 출력, 또는 리디렉션된 출력CRLF에 를 추가할 수 있습니다.

`s3 cp` 명령은 다음 구문을 사용하여 `stdin`에서 지정된 버킷으로 파일 스트림을 업로드합니다.

구문

```
$ aws s3 cp - <target> [--options]
```

`s3 cp` 명령은 다음 구문을 사용하여 `stdout`에 대한 Amazon S3 파일 스트림을 다운로드합니다.

구문

```
$ aws s3 cp <target> [--options] -
```

이 명령과 함께 사용할 몇 가지 일반적인 옵션 및 예제는 s3 명령에 자주 사용되는 옵션 섹션을 참조하세요. 전체 옵션 목록은 AWS CLI 명령 참조에서 `s3 cp` 단원을 참조하세요.

**`s3 cp` 예제**

다음 예제에서는 `s3://amzn-s3-demo-bucket/example`에서 `s3://amzn-s3-demo-bucket/`으로 모든 객체를 복사합니다.

```
$ aws s3 cp s3://amzn-s3-demo-bucket/example s3://amzn-s3-demo-bucket/
```

다음 예제에서는 `s3 cp` 명령을 사용하여 현재 작업 디렉터리에서 Amazon S3 버킷으로 로컬 파일을 복사합니다.

```
$ aws s3 cp filename.txt s3://amzn-s3-demo-bucket
```

다음 예제에서는 Amazon S3 버킷에서 현재 작업 디렉터리로 파일을 복사합니다. 여기서 `./`는 현재 작업 디렉터리를 지정합니다.

```
$ aws s3 cp s3://amzn-s3-demo-bucket/filename.txt ./
```

다음 예제에서는 echo를 사용하여 "hello world" 텍스트를 s3://bucket-name/filename.txt 파일로 스트리밍합니다.

```
$ echo "hello world" | aws s3 cp - s3://amzn-s3-demo-bucket/filename.txt
```

다음 예제에서는 s3://amzn-s3-demo-bucket/filename.txt 파일을 stdout으로 스트리밍하고 내용을 콘솔로 인쇄합니다.

```
$ aws s3 cp s3://amzn-s3-demo-bucket/filename.txt -
hello world
```

다음 예제에서는 s3://bucket-name/pre의 내용을 stdout으로 스트리밍하고, bzip2 명령을 사용하여 파일을 압축하고 key.bz2라는 새 압축 파일을 s3://bucket-name에 업로드합니다.

```
$ aws s3 cp s3://amzn-s3-demo-bucket/pre - | bzip2 --best | aws s3 cp - s3://amzn-s3-demo-bucket/key.bz2
```

객체 동기화

s3 sync 명령은 버킷과 디렉터리의 콘텐츠 또는 두 버킷의 콘텐츠를 동기화합니다. 일반적으로 s3 sync는 원본과 대상 간에 누락되거나 오래된 파일 또는 객체를 복사합니다. 하지만 --delete 옵션을 제공하여 원본에 없는 파일이나 객체를 대상에서 제거할 수도 있습니다.

구문

```
$ aws s3 sync <source> <target> [--options]
```

이 명령과 함께 사용할 몇 가지 일반적인 옵션 및 예제는 s3 명령에 자주 사용되는 옵션 섹션을 참조하세요. 전체 옵션 목록은 AWS CLI 명령 참조에서 s3 sync 단원을 참조하세요.

s3 동기화 예제

다음 예제에서는 amzn-s3-demo-bucket이라는 버킷의 경로라는 Amazon S3 접두사의 내용을 현재 작업 디렉터리와 동기화합니다.

s3 sync는 대상에 있는 동일한 이름의 파일과 크기 또는 수정 시간이 다른 모든 파일을 업데이트합니다. 출력에는 동기화 중에 수행된 특정 작업이 표시됩니다. 이 작업은 하위 디렉터리

MySubdirectory와 해당 내용을 s3://amzn-s3-demo-bucket/path/MySubdirectory와 반복적으로 동기화합니다.

```
$ aws s3 sync . s3://mamzn-s3-demo-bucket/path
upload: MySubdirectory\MyFile3.txt to s3://amzn-s3-demo-bucket/path/MySubdirectory/
MyFile3.txt
upload: MyFile2.txt to s3://amzn-s3-demo-bucket/path/MyFile2.txt
upload: MyFile1.txt to s3://amzn-s3-demo-bucket/path/MyFile1.txt
```

다음 예제에서는 이전 예제를 확장하여 --delete 옵션을 사용하는 방법을 보여줍니다.

```
// Delete local file
$ rm ./MyFile1.txt

// Attempt sync without --delete option - nothing happens
$ aws s3 sync . s3://amzn-s3-demo-bucket/path

// Sync with deletion - object is deleted from bucket
$ aws s3 sync . s3://amzn-s3-demo-bucket/path --delete
delete: s3://amzn-s3-demo-bucket/path/MyFile1.txt

// Delete object from bucket
$ aws s3 rm s3://amzn-s3-demo-bucket/path/MySubdirectory/MyFile3.txt
delete: s3://amzn-s3-demo-bucket/path/MySubdirectory/MyFile3.txt

// Sync with deletion - local file is deleted
$ aws s3 sync s3://amzn-s3-demo-bucket/path . --delete
delete: MySubdirectory\MyFile3.txt

// Sync with Infrequent Access storage class
$ aws s3 sync . s3://amzn-s3-demo-bucket/path --storage-class STANDARD_IA
```

--delete 옵션을 사용할 때 --exclude 및 --include 옵션은 s3 sync 작업 중에 삭제할 파일 또는 객체를 필터링할 수 있습니다. 이 경우 파라미터 문자열은 대상 디렉터리 또는 버킷의 맥락에서 삭제에서 제외하거나 삭제를 위해 포함할 파일을 지정해야 합니다. 다음은 그 한 예입니다.

```
Assume local directory and s3://amzn-s3-demo-bucket/path currently in sync and each
 contains 3 files:
MyFile1.txt
MyFile2.rtf
MyFile88.txt
'''
```

```
// Sync with delete, excluding files that match a pattern. MyFile88.txt is deleted,
 while remote MyFile1.txt is not.
$ aws s3 sync . s3://amzn-s3-demo-bucket/path --delete --exclude "path/MyFile?.txt"
delete: s3://amzn-s3-demo-bucket/path/MyFile88.txt
'''

// Sync with delete, excluding MyFile2.rtf - local file is NOT deleted
$ aws s3 sync s3://amzn-s3-demo-bucket/path . --delete --exclude "./MyFile2.rtf"
download: s3://amzn-s3-demo-bucket/path/MyFile1.txt to MyFile1.txt
'''

// Sync with delete, local copy of MyFile2.rtf is deleted
$ aws s3 sync s3://amzn-s3-demo-bucket/path . --delete
delete: MyFile2.rtf
```

### s3 명령에 자주 사용되는 옵션

다음 옵션은 이 주제에서 설명하는 명령에 자주 사용됩니다. 명령에 사용할 수 있는 옵션의 전체 목록은 AWS CLI 참조 가이드 의 특정 명령을 참조하세요.

#### acl

`s3 sync` 및 `s3 cp`는 `--acl` 옵션을 사용할 수 있습니다. 이렇게 하면 Amazon S3에 복사된 파일에 대한 액세스 권한을 설정할 수 있습니다. `--acl` 옵션에는 `private`, `public-read` 및 `public-read-write` 값을 적용할 수 있습니다. 자세한 내용은 Amazon S3 사용 설명서의 사전 준비를 ACL 참조하세요.

```
$ aws s3 sync . s3://amzn-s3-demo-bucket/path --acl public-read
```

#### exclude

`s3 cp`, `s3 mv`, `s3 sync` 또는 `s3 rm` 명령을 사용하는 경우 `--exclude` 또는 `--include` 옵션을 사용하여 결과를 필터링할 수 있습니다. `--exclude` 옵션은 명령에서 객체만 제외하도록 규칙을 설정하고 옵션은 지정된 순서대로 적용됩니다. 방법은 다음 예제와 같습니다.

```
Local directory contains 3 files:
MyFile1.txt
MyFile2.rtf
MyFile88.txt

// Exclude all .txt files, resulting in only MyFile2.rtf being copied
```

```
$ aws s3 cp . s3://amzn-s3-demo-bucket/path --exclude "*.txt"

// Exclude all .txt files but include all files with the "MyFile*.txt" format,
 resulting in, MyFile1.txt, MyFile2.rtf, MyFile88.txt being copied
$ aws s3 cp . s3://amzn-s3-demo-bucket/path --exclude "*.txt" --include
 "MyFile*.txt"

// Exclude all .txt files, but include all files with the "MyFile*.txt" format,
 but exclude all files with the "MyFile?.txt" format resulting in, MyFile2.rtf and
 MyFile88.txt being copied
$ aws s3 cp . s3://amzn-s3-demo-bucket/path --exclude "*.txt" --include
 "MyFile*.txt" --exclude "MyFile?.txt"
```

포함

s3 cp, s3 mv, s3 sync 또는 s3 rm 명령을 사용하는 경우 --exclude 또는 --include 옵션을 사용하여 결과를 필터링할 수 있습니다. --include 옵션은 명령에 지정된 객체만 포함하도록 규칙을 설정하며 옵션은 지정된 순서대로 적용됩니다. 방법은 다음 예제와 같습니다.

```
Local directory contains 3 files:
MyFile1.txt
MyFile2.rtf
MyFile88.txt

// Include all .txt files, resulting in MyFile1.txt and MyFile88.txt being copied
$ aws s3 cp . s3://amzn-s3-demo-bucket/path --include "*.txt"

// Include all .txt files but exclude all files with the "MyFile*.txt" format,
 resulting in no files being copied
$ aws s3 cp . s3://amzn-s3-demo-bucket/path --include "*.txt" --exclude
 "MyFile*.txt"

// Include all .txt files, but exclude all files with the "MyFile*.txt" format, but
 include all files with the "MyFile?.txt" format resulting in MyFile1.txt being
 copied

$ aws s3 cp . s3://amzn-s3-demo-bucket/path --include "*.txt" --exclude
 "MyFile*.txt" --include "MyFile?.txt"
```

권한 부여

s3 cp, s3 mv 및 s3 sync 명령에는 지정된 사용자 또는 그룹에게 객체에 대한 권한을 부여하기 위해 사용할 수 있는 --grants 옵션이 포함됩니다. 다음 구문을 사용하여 --grants 옵션을 권한

목록으로 설정합니다. Permission, Grantee_Type 및 Grantee_ID를 사용자의 값으로 바꿉니다.

구문

```
--grants Permission=Grantee_Type=Grantee_ID
         [Permission=Grantee_Type=Grantee_ID ...]
```

각 값에는 다음 요소가 포함됩니다.

- *Permission* - 부여된 권한을 지정합니다. read, readacl, writeacl 또는 full로 설정할 수 있습니다.
- *Grantee_Type* - 권한 부여자를 식별하는 방법을 지정합니다. uri, emailaddress 또는 id로 설정할 수 있습니다.
- *Grantee_ID* - 다음을 기반으로 권한 부여자를 지정합니다.*Grantee_Type*.
  - uri - 그룹의 입니다URI. 자세한 내용은 피부여자란?을 참조하세요.
  - emailaddress - 계정의 이메일 주소입니다.
  - id - 계정의 정식 ID입니다.

Amazon S3 액세스 제어에 대한 자세한 내용은 액세스 제어를 참조하세요.

다음 예제에서는 버킷에 객체를 복사합니다. 여기서는 모든 사람에게 객체에 대한 read 권한을 부여하고 full과 연결된 계정에 read 권한(readacl, writeacl 및 user@example.com)을 부여합니다.

```
$ aws s3 cp file.txt s3://amzn-s3-demo-bucket/ --grants read=uri=http://
acs.amazonaws.com/groups/global/AllUsers full=emailaddress=user@example.com
```

Amazon S3에 업로드하는 객체에 대해 기본값이 아닌 스토리지 클래스(REDUCED_REDUNDANCY 또는 STANDARD_IA)를 지정할 수도 있습니다. 이렇게 하려면 --storage-class 옵션을 사용합니다.

```
$ aws s3 cp file.txt s3://amzn-s3-demo-bucket/ --storage-class REDUCED_REDUNDANCY
```

recursive

이 옵션을 사용하면 지정된 디렉터리 또는 접두사 아래의 모든 파일 또는 객체에 대해 명령이 수행됩니다. 다음 예제에서는 s3://amzn-s3-demo-bucket/path 및 모든 내용을 삭제합니다.

```
$ aws s3 rm s3://amzn-s3-demo-bucket/path --recursive
```

리소스

AWS CLI 참조:

- aws s3
- aws s3 cp
- aws s3 mb
- aws s3 mv
- aws s3 ls
- aws s3 rb
- aws s3 rm
- aws s3 sync

서비스 참조:

- Amazon S3 사용 설명서의 Amazon S3 버킷 작업 Amazon S3
- Amazon S3 사용 설명서의 Amazon S3 객체 작업 Amazon S3
- Amazon S3 사용 설명서의 접두사와 구분자를 사용하여 계층적으로 키 나열
- Amazon S3 사용 설명서의 AWS SDK for .NET (하위 수준)을 사용하여 S3 버킷에 대한 멀티파트 업로드를 중단합니다. Amazon S3

## 에서 API-Level(s3api) 명령 사용 AWS CLI

API-레벨 명령(s3api명령 세트에 포함됨)은 Amazon Simple Storage Service(Amazon S3) 에 대한 직접 액세스를 제공하고 상위 레벨 s3 명령에 노출되지 않는 일부 작업을 APIs활성화합니다. 이러한 명령은 서비스 기능에 대한 API수준 액세스를 제공하는 다른 AWS 서비스와 동일합니다. s3 명령어에 대한 자세한 내용은 에서 상위 수준 (s3) 명령 사용 AWS CLI 섹션을 참조하세요.

이 주제에서는 Amazon S3 에 매핑되는 하위 수준 명령을 사용하는 방법을 보여주는 예제를 제공합니다APIs. 또한 AWS CLI 참조 가이드 버전 2 참조 가이드 의 s3api 섹션에서 각 S3 API 명령에 대한 예제를 찾을 수 있습니다.

주제



## 사전 조건

`s3api` 명령을 실행하려면 다음을 수행해야 합니다.

- AWS CLI를 설치하고 구성합니다. 자세한 내용은 설치 AWS CLI 및 에 대한 인증 및 액세스 자격 증명 AWS CLI 단원을 참조하세요.
- 사용하는 프로필에는 예제에서 수행하는 AWS 작업을 허용하는 권한이 있어야 합니다.
- 다음 Amazon S3 용어를 이해하세요.
  - 버킷 - 최상위 Amazon S3 폴더입니다.
  - 접두사 - 버킷의 Amazon S3 폴더입니다.
  - 객체 - Amazon S3 버킷에서 호스팅되는 모든 항목입니다.

## 사용자 지정 적용 ACL

상위 수준 명령을 사용하면 `--acl` 옵션을 사용하여 사전 정의된 액세스 제어 목록(ACLs)을 Amazon S3 객체에 적용할 수 있습니다. 하지만 이 명령을 사용하여 버킷 전체 를 설정할 수는 없습니다ACLs. 그러나 `put-bucket-acl` API-level 명령을 사용하여 이 작업을 수행할 수 있습니다.

다음 예제에서는 두 AWS 사용자(user1@example.com 및 user2@example.com)에게 전체 제어 권한을 부여하고 모든 사용자에게 읽기 권한을 부여하는 방법을 보여줍니다. '모두'의 식별자는 파라미터로 URI 전달하는 특수 에서 가져옵니다.

```
$ aws s3api put-bucket-acl --bucket amzn-s3-demo-bucket --grant-full-control
 'emailaddress="user1@example.com",emailaddress="user2@example.com"' --grant-read
 'uri="http://acs.amazonaws.com/groups/global/AllUsers"'
```

를 구성하는 방법에 대한 자세한 ACLs내용은 Amazon Simple Storage Service API 참조의 PUT 버킷 acl을 참조하세요. CLI와 같은 의 `s3api` ACL 명령은 동일한 단축 인수 표기법을 `put-bucket-acl`사용합니다.

로깅 정책 구성

API 명령은 버킷 로깅 정책을 put-bucket-logging 구성합니다.

다음 예제에서는 AWS 사용자 user@example.com에 로그 파일을 완전히 제어할 수 있으며 모든 사용자에게 읽기 액세스 권한이 부여됩니다. 또한 put-bucket-acl 명령은 Amazon S3 로그 전송 시스템(에 의해 지정됨URI)에 버킷에 로그를 읽고 쓰는 데 필요한 권한을 부여하는 데 필요합니다.

```
$ aws s3api put-bucket-acl --bucket amzn-s3-demo-bucket --grant-read-acp 'URI="http://
acs.amazonaws.com/groups/s3/LogDelivery"' --grant-write 'URI="http://acs.amazonaws.com/
groups/s3/LogDelivery"'
$ aws s3api put-bucket-logging --bucket amzn-s3-demo-bucket --bucket-logging-status
 file://logging.json
```

이전 명령의 logging.json 파일에도 다음 내용이 있습니다.

```
{
  "LoggingEnabled": {
    "TargetBucket": "amzn-s3-demo-bucket",
    "TargetPrefix": "amzn-s3-demo-bucketLogs/",
    "TargetGrants": [
      {
        "Grantee": {
          "Type": "AmazonCustomerByEmail",
          "EmailAddress": "user@example.com"
        },
        "Permission": "FULL_CONTROL"
      },
      {
        "Grantee": {
          "Type": "Group",
          "URI": "http://acs.amazonaws.com/groups/global/AllUsers"
        },
        "Permission": "READ"
      }
    ]
  }
}
```

리소스

AWS CLI 참조:

- `aws s3api`
- `aws s3api put-bucket-acl`
- `aws s3api put-bucket-logging`

서비스 참조:

- Amazon S3 사용 설명서의 Amazon S3 버킷 작업 Amazon S3
- Amazon S3 사용 설명서의 Amazon S3 객체 작업 Amazon S3
- Amazon S3 사용 설명서의 접두사 및 구분자를 사용하여 계층적으로 키 나열
- Amazon S3 사용 설명서의 AWS SDK for .NET (하위 수준)를 사용하여 S3 버킷에 대한 멀티파트 업로드를 중단합니다. Amazon S3

## 의 Amazon S3 버킷 수명 주기에 대한 스크립팅 예제 AWS CLI

이 주제에서는 AWS Command Line Interface (AWS CLI)를 사용하는 Amazon S3 버킷 수명 주기 작업에 대한 bash 스크립팅 예제를 사용합니다. 이 스크립팅 예제에서는 `aws s3api` 명령 세트를 사용합니다. 셸 스크립트는 명령줄 인터페이스에서 실행되도록 설계된 프로그램입니다.

주제

- 시작하기 전에
- 이 예제 정보
- 파일
- 참조

시작하기 전에

아래 예제 중 하나를 실행하려면 먼저 다음 작업을 완료해야 합니다.

- AWS CLI를 설치하고 구성합니다. 자세한 내용은 설치 AWS CLI 및 에 대한 인증 및 액세스 자격 증명 AWS CLI 단원을 참조하세요.
- 사용하는 프로필에는 예제에서 수행하는 AWS 작업을 허용하는 권한이 있어야 합니다.

- AWS 가장 좋은 방법은 이 코드에 최소 권한을 부여하거나 작업을 수행하는 데 필요한 권한만 부여하는 것입니다. 자세한 내용은 IAM 사용 설명서의 최소 권한 부여를 참조하세요.

- 이 코드는 일부 AWS 리전에서 테스트되지 않았습니다. 일부 AWS 서비스는 특정 리전에서만 사용할 수 있습니다. 자세한 내용은 AWS 일반 참조 안내서에서 서비스 엔드포인트 및 할당량을 참조하세요.
- 이 코드를 실행하면 AWS 계정에 요금이 부과될 수 있습니다. 이 스크립트에 의해 생성된 모든 리소스를 사용한 후 제거하는 것은 사용자의 책임입니다.

Amazon S3 서비스에는 다음 용어가 사용됩니다.

- 버킷 - 최상위 수준의 Amazon S3 폴더입니다.
- 접두사 - 버킷의 Amazon S3 폴더입니다.
- 객체 - Amazon S3 버킷에서 호스팅되는 모든 항목입니다.

이 예제 정보

이 예제는 셸 스크립트 파일의 함수 세트를 사용하여 일부 기본 Amazon S3 작업과 상호 작용하는 방법을 보여줍니다. 함수는 `bucket-operations.sh`라는 셸 스크립트 파일에 있습니다. 다른 파일에서 이러한 함수를 호출할 수 있습니다. 각 스크립트 파일에는 각 함수를 설명하는 주석이 들어 있습니다.

각 단계의 중간 결과를 보려면 `-i` 파라미터와 함께 스크립트를 실행합니다. Amazon S3 콘솔을 사용하여 버킷의 현재 상태 또는 버킷의 내용을 볼 수 있습니다. 스크립트는 프롬프트에서 Enter 키를 누르는 경우에만 다음 단계로 진행됩니다.

전체 예제 및 다운로드 가능한 스크립트 파일은 의 AWS 코드 예제 리포지토리에서 Amazon S3 버킷 수명 주기 작업을 참조하세요GitHub.

파일

에제에는 다음 파일이 들어 있습니다.

bucket-operations.sh

이 기본 스크립트 파일은 다른 파일에서 가져올 수 있습니다. 여기에는 다음 작업을 수행하는 함수가 포함됩니다.

- 버킷 생성 및 버킷이 존재하는지 확인
- 로컬 컴퓨터에서 버킷으로 파일 복사
- 한 버킷 위치에서 다른 버킷 위치로 파일 복사

- 버킷의 내용 나열
- 버킷에서 파일 삭제
- 버킷 삭제

에서 bucket-operations.sh 에 대한 코드를 확인합니다GitHub.

test-bucket-operations.sh

셸 스크립트 파일 test-bucket-operations.sh는 bucket-operations.sh 파일을 소싱하고 각 함수를 호출하여 함수를 호출하는 방법을 보여줍니다. 함수를 호출한 후 테스트 스크립트는 생성한 모든 리소스를 제거합니다.

에서 test-bucket-operations.sh 에 대한 코드를 확인합니다GitHub.

awsdocs-general.sh

스크립트 파일 awsdocs-general.sh에는 AWS CLI에 대한 고급 코드 예제에서 사용되는 범용 함수가 들어 있습니다.

에서 awsdocs-general.sh 에 대한 코드를 확인합니다GitHub.

참조

AWS CLI 참조:

- 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

기타 참조:

- Amazon S3 사용 설명서의 Amazon S3 버킷 작업 Amazon S3
- Amazon S3 사용 설명서의 Amazon S3 객체 작업 Amazon S3
- 및 AWS CLI 코드 예제를 AWS SDK 보고 기여하려면 의 AWS 코드 예제 리포지토리를 참조하세요GitHub.

## SNS 에서 Amazon 액세스 AWS CLI

AWS Command Line Interface (SNS)를 사용하여 Amazon Simple Notification Service(Amazon)의 기능에 액세스할 수 있습니다AWS CLI. Amazon 에 대한 AWS CLI 명령을 나열하려면 다음 명령을 SNS 사용합니다.

```
aws sns help
```

명령을 실행하기 전에 기본 자격 증명을 설정합니다. 자세한 내용은 에 대한 설정 구성 AWS CLI 단원을 참조하십시오.

이 주제에서는 Amazon 에 대한 일반적인 작업을 수행하는 AWS CLI 명령의 예를 보여줍니다SNS.

주제

- 주제 생성
- 주제 구독
- 주제 게시
- 주제에서 구독 취소
- 주제 삭제

### 주제 생성

주제를 만들려면 `sns create-topic` 명령을 사용하고 주제에 할당할 이름을 지정합니다.

```
$ aws sns create-topic --name my-topic
{
    "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-topic"
}
```

나중에 메시지를 게시할 때 사용할 응답의 `TopicArn`을 적어 둡니다.

## 주제 구독

주제를 구독하려면 sns subscribe 명령을 사용합니다.

다음 예제는 email에 대한 이메일 주소와 notification-endpoint 프로토콜을 지정합니다.

```
$ aws sns subscribe --topic-arn arn:aws:sns:us-west-2:123456789012:my-topic --
protocol email --notification-endpoint saanvi@example.com
{
    "SubscriptionArn": "pending confirmation"
}
```

AWS 는 subscribe 명령에 지정한 주소로 이메일을 통해 확인 메시지를 즉시 보냅니다. 이메일 메시지에는 다음 텍스트가 포함됩니다.

```
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
```

수신자가 구독 확인 링크를 클릭하면 수신자의 브라우저에 다음과 유사한 정보가 포함된 알림 메시지가 표시됩니다.

```
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.
```

## 주제 게시

주제의 모든 구독자에게 메시지를 보내려면 sns publish 명령을 사용합니다.

다음 예제에서는 지정된 주제의 모든 가입자에게 'Hello World!'라는 메시지를 보냅니다.

```
$ aws sns publish --topic-arn arn:aws:sns:us-west-2:123456789012:my-topic --
message "Hello World!"
{
```

```
    "MessageId": "4e41661d-5eec-5ddf-8dab-2c867EXAMPLE"
}
```

이 예제에서는 'Hello World!'라는 텍스트가 포함된 이메일 메시지를 AWS 보냅니다. `saanvi@example.com`으로 전송합니다.

### 주제에서 구독 취소

주제를 구독 취소하고 해당 주제에 게시된 메시지 수신을 중지하려면 `sns unsubscribe` 명령을 사용하고 구독 취소하려는 주제ARN의 를 지정합니다.

```
$ aws sns unsubscribe --subscription-arn arn:aws:sns:us-west-2:123456789012:my-topic:1328f057-de93-4c15-512e-8bb22EXAMPLE
```

구독을 성공적으로 취소했는지 확인하려면 `sns list-subscriptions` 명령을 사용하여 가 목록에 더 이상 나타나지 ARN 않는지 확인합니다.

```
$ aws sns list-subscriptions
```

### 주제 삭제

주제를 삭제하려면 `sns delete-topic` 명령을 실행합니다.

```
$ aws sns delete-topic --topic-arn arn:aws:sns:us-west-2:123456789012:my-topic
```

주제를 AWS 성공적으로 삭제했는지 확인하려면 `sns list-topics` 명령을 사용하여 주제가 목록에 더 이상 나타나지 않는지 확인합니다.

```
$ aws sns list-topics
```

## AWS CLI 명령 예제

이 주제의 코드 예제는 와 AWS Command Line Interface 함께 를 사용하는 방법을 보여줍니다 AWS.

기본 사항은 서비스 내에서 필수 작업을 수행하는 방법을 보여주는 코드 예제입니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

시나리오는 동일한 서비스 내에서 또는 다른 AWS 서비스와 결합된 상태에서 여러 함수를 호출하여 특정 태스크를 수행하는 방법을 보여주는 코드 예제입니다.

서비스

- ACM 사용 예제 AWS CLI
- API 를 사용한 게이트웨이 예제 AWS CLI
- API 를 사용한 게이트웨이 HTTP 및 WebSocket API 예제 AWS CLI
- API 를 사용한 Gateway Management API 예제 AWS CLI
- 를 사용한 App Mesh 예제 AWS CLI
- 를 사용한 App Runner 예제 AWS CLI
- AWS AppConfig 사용 예제 AWS CLI
- 를 사용한 Application Auto Scaling 예제 AWS CLI
- 를 사용한 Application Discovery Service 예제 AWS CLI
- AppRegistry 사용 예제 AWS CLI
- 를 사용한 Athena 예제 AWS CLI
- 를 사용한 Auto Scaling 예제 AWS CLI
- 를 사용한 Auto Scaling Plans 예제 AWS CLI
- AWS Backup 사용 예제 AWS CLI
- AWS Batch 사용 예제 AWS CLI
- AWS Budgets 사용 예제 AWS CLI
- 를 사용한 Amazon Chime 예제 AWS CLI
- 를 사용한 Cloud Control API 예제 AWS CLI
- AWS Cloud Map 사용 예제 AWS CLI
- AWS Cloud9 사용 예제 AWS CLI
- AWS CloudFormation 사용 예제 AWS CLI
- CloudFront 사용 예제 AWS CLI
- 를 사용한 Amazon CloudSearch 예제 AWS CLI
- CloudTrail 사용 예제 AWS CLI
- CloudWatch 사용 예제 AWS CLI
- CloudWatch 를 사용하여 예제를 로깅합니다. AWS CLI

- CloudWatch 를 사용한 네트워크 모니터링 예제 AWS CLI
- CodeArtifact 사용 예제 AWS CLI
- CodeBuild 사용 예제 AWS CLI
- CodeCommit 사용 예제 AWS CLI
- CodeDeploy 사용 예제 AWS CLI
- CodeGuru 를 사용한 검토자 예제 AWS CLI
- CodePipeline 사용 예제 AWS CLI
- AWS CodeStar 를 사용한 알림 예제 AWS CLI
- CodeConnections 사용 예제 AWS CLI
- 를 사용한 Amazon Cognito 자격 증명 예제 AWS CLI
- 를 사용한 Amazon Cognito Identity Provider 예제 AWS CLI
- 를 사용한 Amazon Comprehend 예제 AWS CLI
- 를 사용한 Amazon Comprehend Medical 예제 AWS CLI
- AWS Config 사용 예제 AWS CLI
- 를 사용한 Amazon Connect 예제 AWS CLI
- AWS Cost and Usage Report 사용 예제 AWS CLI
- 를 사용한 Cost Explorer 서비스 예제 AWS CLI
- 를 사용한 Firehose 예제 AWS CLI
- 를 사용한 Amazon Data Lifecycle Manager 예제 AWS CLI
- AWS Data Pipeline 사용 예제 AWS CLI
- DataSync 사용 예제 AWS CLI
- DAX 사용 예제 AWS CLI
- 를 사용한 Detective 예제 AWS CLI
- 를 사용한 Device Farm 예제 AWS CLI
- AWS Direct Connect 사용 예제 AWS CLI
- AWS Directory Service 사용 예제 AWS CLI
- AWS DMS 사용 예제 AWS CLI
- 를 사용한 Amazon DocumentDB 예제 AWS CLI

- 를 사용한 DynamoDB 예제 AWS CLI
- 를 사용한 DynamoDB Streams 예제 AWS CLI
- 를 사용한 Amazon EC2 예제 AWS CLI
- 를 사용한 Amazon EC2 Instance Connect 예제 AWS CLI
- 를 사용한 Amazon ECR 예제 AWS CLI
- 를 사용한 Amazon ECR Public 예제 AWS CLI
- 를 사용한 Amazon ECS 예제 AWS CLI
- 를 사용한 Amazon EFS 예제 AWS CLI
- 를 사용한 Amazon EKS 예제 AWS CLI
- 를 사용한 Elastic Beanstalk 예제 AWS CLI
- Elastic Load Balancing - 를 사용하는 버전 1 예제 AWS CLI
- Elastic Load Balancing - 를 사용한 버전 2 예제 AWS CLI
- 를 사용한 Elastic Transcoder 예제 AWS CLI
- ElastiCache 사용 예제 AWS CLI
- MediaStore 사용 예제 AWS CLI
- 를 사용한 Amazon EMR 예제 AWS CLI
- 를 사용한 Amazon EMR on EKS 예제 AWS CLI
- EventBridge 사용 예제 AWS CLI
- 를 사용한 Firewall Manager 예제 AWS CLI
- AWS FIS 사용 예제 AWS CLI
- 를 사용한 Amazon GameLift 예제 AWS CLI
- 를 사용한 Global Accelerator 예제 AWS CLI
- AWS Glue 사용 예제 AWS CLI
- GuardDuty 사용 예제 AWS CLI
- AWS Health 사용 예제 AWS CLI
- HealthImaging 사용 예제 AWS CLI
- HealthLake 사용 예제 AWS CLI
- HealthOmics 사용 예제 AWS CLI

- IAM 사용 예제 AWS CLI
- IAM 를 사용하여 분석기 예제 액세스 AWS CLI
- 를 사용한 Image Builder 예제 AWS CLI
- 를 사용한 Incident Manager 예제 AWS CLI
- 를 사용한 Incident Manager Contacts 예제 AWS CLI
- 를 사용한 Amazon Inspector 예제 AWS CLI
- AWS IoT 사용 예제 AWS CLI
- AWS IoT 1-Click 를 사용한 디바이스 예제 AWS CLI
- AWS IoT 1-Click 를 사용한 프로젝트 예제 AWS CLI
- AWS IoT Analytics 사용 예제 AWS CLI
- 를 사용한 Device Advisor 예제 AWS CLI
- AWS IoT data 사용 예제 AWS CLI
- AWS IoT Events 사용 예제 AWS CLI
- AWS IoT Events-Data 사용 예제 AWS CLI
- AWS IoT Greengrass 사용 예제 AWS CLI
- AWS IoT Greengrass V2 사용 예제 AWS CLI
- AWS IoT Jobs SDK release 사용 예제 AWS CLI
- AWS IoT SiteWise 사용 예제 AWS CLI
- AWS IoT Things Graph 사용 예제 AWS CLI
- AWS IoT 무선 사용 예제 AWS CLI
- 를 사용한 Amazon IVS 예제 AWS CLI
- 를 사용한 Amazon IVS Chat 예제 AWS CLI
- 를 사용한 Amazon IVS 실시간 스트리밍 예제 AWS CLI
- 를 사용한 Amazon Kendra 예제 AWS CLI
- 를 사용한 Kinesis 예제 AWS CLI
- AWS KMS 사용 예제 AWS CLI
- 를 사용한 Lake Formation 예제 AWS CLI
- 를 사용한 Lambda 예제 AWS CLI
- 를 사용한 License Manager 예제 AWS CLI

- 를 사용한 Lightsail 예제 AWS CLI
- 를 사용한 Macie 예제 AWS CLI
- 를 사용한 Amazon Managed Grafana 예제 AWS CLI
- MediaConnect 사용 예제 AWS CLI
- MediaConvert 사용 예제 AWS CLI
- MediaLive 사용 예제 AWS CLI
- MediaPackage 사용 예제 AWS CLI
- MediaPackage VOD 사용 예제 AWS CLI
- MediaStore 를 사용한 데이터 플레인 예제 AWS CLI
- MediaTailor 사용 예제 AWS CLI
- 를 사용한 MemoryDB 예제 AWS CLI
- 를 사용한 Amazon MSK 예제 AWS CLI
- 를 사용한 Network Manager 예제 AWS CLI
- 를 사용한 Nimble Studio 예제 AWS CLI
- OpenSearch 를 사용한 서비스 예제 AWS CLI
- AWS OpsWorks 사용 예제 AWS CLI
- AWS OpsWorks CM 사용 예제 AWS CLI
- 를 사용한 조직 예제 AWS CLI
- AWS Outposts 사용 예제 AWS CLI
- AWS Payment Cryptography 사용 예제 AWS CLI
- AWS Payment Cryptography 를 사용한 데이터 플레인 예제 AWS CLI
- 를 사용한 Amazon Pinpoint 예제 AWS CLI
- 를 사용한 Amazon Polly 예제 AWS CLI
- AWS 가격표 사용 예제 AWS CLI
- AWS Private CA 사용 예제 AWS CLI
- AWS Proton 사용 예제 AWS CLI
- QLDB 사용 예제 AWS CLI
- 를 사용한 Amazon RDS 예제 AWS CLI
- 를 사용한 Amazon RDS Data Service 예제 AWS CLI

- 를 사용한 Amazon RDS 성능 개선 도우미 예제 AWS CLI
- 를 사용한 Amazon Redshift 예제 AWS CLI
- 를 사용한 Amazon Rekognition 예제 AWS CLI
- AWS RAM 사용 예제 AWS CLI
- 를 사용한 Resource Explorer 예제 AWS CLI
- 를 사용한 리소스 그룹 예제 AWS CLI
- 를 사용하여 리소스 그룹 태그 지정 API 예제 AWS CLI
- AWS RoboMaker 사용 예제 AWS CLI
- 를 사용하여 Route 53 예제 AWS CLI
- 를 사용하여 Route 53 도메인 등록 예제 AWS CLI
- 를 사용하여 Route 53 Profiles 예제 AWS CLI
- 를 사용하여 Route 53 Resolver 예제 AWS CLI
- 를 사용한 Amazon S3 예제 AWS CLI
- 를 사용한 Amazon S3 컨트롤 예제 AWS CLI
- 를 사용한 S3 Glacier 예제 AWS CLI
- 를 사용한 Secrets Manager 예제 AWS CLI
- 를 사용한 Security Hub 예제 AWS CLI
- 를 사용한 Security Lake 예제 AWS CLI
- AWS Serverless Application Repository 사용 예제 AWS CLI
- 를 사용한 서비스 카탈로그 예제 AWS CLI
- 를 사용한 Service Quotas 예제 AWS CLI
- 를 사용한 Amazon SES 예제 AWS CLI
- 를 사용하여 예제 보호 AWS CLI
- 를 사용한 서명자 예제 AWS CLI
- 를 사용한 Snowball 예제 AWS CLI
- 를 사용한 Amazon SNS 예제 AWS CLI
- 를 사용한 Amazon SQS 예제 AWS CLI
- 를 사용한 Storage Gateway 예제 AWS CLI
- AWS STS 사용 예제 AWS CLI

- AWS Support 사용 예제 AWS CLI
- 를 사용한 Amazon SWF 예제 AWS CLI
- 를 사용한 Systems Manager 예제 AWS CLI
- 를 사용한 Amazon Textract 예제 AWS CLI
- 를 사용한 Amazon Transcribe 예제 AWS CLI
- 를 사용한 Amazon Translate 예제 AWS CLI
- Trusted Advisor 사용 예제 AWS CLI
- 를 사용하여 확인된 권한 예제 AWS CLI
- VPC 를 사용한 Lattice 예제 AWS CLI
- AWS WAF Classic 사용 예제 AWS CLI
- AWS WAF Classic Regional 사용 예제 AWS CLI
- AWS WAFV2 사용 예제 AWS CLI
- 를 사용한 Amazon WorkDocs 예제 AWS CLI
- 를 사용한 Amazon WorkMail 예제 AWS CLI
- 를 사용한 Amazon WorkMail 메시지 흐름 예제 AWS CLI
- WorkSpaces 사용 예제 AWS CLI
- 를 사용한 X-Ray 예제 AWS CLI

## ACM 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다ACM.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### add-tags-to-certificate

다음 코드 예시에서는 add-tags-to-certificate을 사용하는 방법을 보여 줍니다.

AWS CLI

기존 ACM 인증서에 태그를 추가하려면

다음 add-tags-to-certificate 명령은 지정된 인증서에 두 개의 태그를 추가합니다. 공백 하나를 사용하여 여러 태그를 구분합니다.

```
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
```

- 자세한 API 내용은 명령 참조AddTagsToCertificate의 섹션을 참조하세요. AWS CLI

### delete-certificate

다음 코드 예시에서는 delete-certificate을 사용하는 방법을 보여 줍니다.

AWS CLI

계정에서 ACM 인증서를 삭제하려면

다음 delete-certificate 명령은 지정된 가 있는 인증서를 삭제합니다ARN.

```
aws acm delete-certificate --certificate-
arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012
```

- 자세한 API 내용은 명령 참조DeleteCertificate의 섹션을 참조하세요. AWS CLI

### describe-certificate

다음 코드 예시에서는 describe-certificate을 사용하는 방법을 보여 줍니다.

AWS CLI

ACM 인증서에 포함된 필드를 검색하려면

다음 describe-certificate 명령은 지정된 를 사용하여 인증서의 모든 필드를 검색합니다 ARN.

```
aws acm describe-certificate --certificate-
arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012
```

다음과 유사한 출력이 표시됩니다.

```
{
  "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"
    ]
  }
}
```

- 자세한 API 내용은 명령 참조DescribeCertificate의 섹션을 참조하세요. AWS CLI

### `export-certificate`

다음 코드 예시에서는 `export-certificate`을 사용하는 방법을 보여 줍니다.

AWS CLI

프라이빗 CA에서 발급한 프라이빗 인증서를 내보냅니다.

다음 `export-certificate` 명령은 프라이빗 인증서, 인증서 체인 및 프라이빗 키를 디스플레이로 내보냅니다.

```
aws acm export-certificate --certificate-
arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012 --
passphrase file://path-to-passphrase-file
```

인증서, 체인 및 프라이빗 키를 로컬 파일로 내보내려면 다음 명령을 사용합니다.

```
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
```

- 자세한 API 내용은 명령 참조ExportCertificate의 섹션을 참조하세요. AWS CLI

### get-certificate

다음 코드 예시에서는 get-certificate을 사용하는 방법을 보여 줍니다.

AWS CLI

ACM 인증서를 검색하려면

다음 get-certificate 명령은 지정된 ARN 및 인증서 체인에 대한 인증서를 검색합니다.

```
aws acm get-certificate --certificate-
arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012
```

다음과 유사한 출력이 표시됩니다.

```
{
  "Certificate": "-----BEGIN CERTIFICATE-----
MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----",

  "CertificateChain": "-----BEGIN CERTIFICATE-----
MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
```

```
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----",
"-----BEGIN CERTIFICATE-----
MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----",
"-----BEGIN CERTIFICATE-----
MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----"
}
```

- 자세한 API 내용은 명령 참조GetCertificate의 섹션을 참조하세요. AWS CLI

### import-certificate

다음 코드 예시에서는 import-certificate을 사용하는 방법을 보여 줍니다.

AWS CLI

인증서를 로 가져옵니다ACM.

다음 import-certificate 명령은 인증서를 로 가져옵니다ACM. 실제 파일 이름으로 바꾸세요.

```
aws acm import-certificate --certificate file://Certificate.pem --certificate-
chain file://CertificateChain.pem --private-key file://PrivateKey.pem
```

- 자세한 API 내용은 명령 참조ImportCertificate의 섹션을 참조하세요. AWS CLI

### list-certificates

다음 코드 예시에서는 list-certificates을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 계정의 ACM 인증서를 나열하려면

다음 list-certificates 명령은 계정의 인증서 ARNs 를 나열합니다.

```
aws acm list-certificates
```

위의 명령은 다음과 비슷한 출력을 생성합니다.

```
{
    "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"
        }
    ]
```

```
}
```

list-certificates를 직접 호출할 때마다 표시할 인증서 수를 결정할 수 있습니다. 예를 들어, 네 개의 인증서가 있고 한 번에 두 개까지만 표시하려는 경우 다음 예와 같이 max-items 인수를 2로 설정합니다.

```
aws acm list-certificates --max-items 2
```

두 개의 인증서ARNs와 NextToken 값이 표시됩니다.

```
"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"
```

계정의 다음 인증서 두 개를 표시하려면 다음 직접 호출에서 이 NextToken 값을 설정하세요.

```
aws acm list-certificates --max-items 2 --next-token 9f4d9f69-275a-41fe-
b58e-2b837bd9ba48
```

certificate-statuses 인수를 사용하여 출력을 필터링할 수 있습니다. 다음 명령은 상태가 PENDING_VALIDATION인 인증서를 표시합니다.

```
aws acm list-certificates --certificate-statuses PENDING_VALIDATION
```

includes 인수를 사용하여 출력을 필터링할 수도 있습니다. 다음 명령은 다음 속성에서 필터링된 인증서를 표시합니다. 표시할 인증서:

```
- 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
```

- 자세한 API 내용은 명령 참조ListCertificates의 섹션을 참조하세요. AWS CLI

### `list-tags-for-certificate`

다음 코드 예시에서는 `list-tags-for-certificate`을 사용하는 방법을 보여 줍니다.

AWS CLI

ACM 인증서에 적용된 태그를 나열하려면

다음 `list-tags-for-certificate` 명령은 계정의 인증서에 적용된 태그를 나열합니다.

```
aws acm list-tags-for-certificate --certificate-
arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012
```

위의 명령은 다음과 비슷한 출력을 생성합니다.

```
{
  "Tags": [
      {
          "Value": "Website",
          "Key": "Purpose"
      },
      {
          "Value": "Alice",
          "Key": "Admin"
      }
  ]
}
```

- 자세한 API 내용은 명령 참조ListTagsForCertificate의 섹션을 참조하세요. AWS CLI

### `remove-tags-from-certificate`

다음 코드 예시에서는 `remove-tags-from-certificate`을 사용하는 방법을 보여 줍니다.

AWS CLI

ACM 인증서에서 태그를 제거하려면

다음 `remove-tags-from-certificate` 명령은 지정된 인증서에서 두 개의 태그를 제거합니다. 공백 하나를 사용하여 여러 태그를 구분합니다.

```
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
```

- 자세한 API 내용은 명령 참조RemoveTagsFromCertificate의 섹션을 참조하세요. AWS CLI

### `request-certificate`

다음 코드 예시에서는 `request-certificate`을 사용하는 방법을 보여 줍니다.

AWS CLI

새 ACM 인증서를 요청하려면

다음 `request-certificate` 명령은 DNS 검증을 사용하여 www.example.com 도메인에 대한 새 인증서를 요청합니다.

```
aws acm request-certificate --domain-name www.example.com --validation-method DNS
```

멱등성 토큰을 입력하여 `request-certificate`에 대한 직접 호출을 구분할 수 있습니다.

```
aws acm request-certificate --domain-name www.example.com --validation-method DNS --
idempotency-token 91adc45q
```

하나 이상의 주체 대체 이름을 입력하여 두 개 이상의 apex 도메인을 보호하는 인증서를 요청할 수 있습니다.

```
aws acm request-certificate --domain-name example.com --validation-method DNS --
idempotency-token 91adc45q --subject-alternative-names www.example.net
```

웹 사이트에 접속하는 데도 사용할 수 있는 대체 이름을 입력할 수 있습니다.

```
aws acm request-certificate --domain-name example.com --validation-method DNS --
idempotency-token 91adc45q --subject-alternative-names www.example.com
```

별표(*)를 와일드카드로 사용하여 동일한 도메인 내의 여러 하위 도메인에 대한 인증서를 생성할 수 있습니다.

```
aws acm request-certificate --domain-name example.com --validation-method DNS --
idempotency-token 91adc45q --subject-alternative-names *.example.com
```

대체 이름을 여러 개 입력할 수도 있습니다.

```
aws acm request-certificate --domain-name example.com --validation-method DNS --
subject-alternative-names b.example.com c.example.com d.example.com
```

검증에 이메일을 사용하는 경우 도메인 검증 옵션을 입력하여 검증 이메일을 보낼 도메인을 지정할 수 있습니다.

```
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
```

다음 명령은 새 인증서를 요청할 때 인증서 투명성 로깅을 옵트아웃합니다.

```
aws acm request-certificate --domain-name www.example.com --validation-method DNS --
options CertificateTransparencyLoggingPreference=DISABLED --idempotency-token 184627
```

- 자세한 API 내용은 명령 참조RequestCertificate의 섹션을 참조하세요. AWS CLI

### resend-validation-email

다음 코드 예시에서는 resend-validation-email을 사용하는 방법을 보여 줍니다.

AWS CLI

ACM 인증서 요청에 대한 검증 이메일을 다시 보내려면

다음 resend-validation-email 명령은 Amazon 인증 기관에 적절한 주소로 검증 이메일을 보내도록 지시합니다.

```
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
```

- 자세한 API 내용은 명령 참조ResendValidationEmail의 섹션을 참조하세요. AWS CLI

**update-certificate-options**

다음 코드 예시에서는 update-certificate-options을 사용하는 방법을 보여 줍니다.

AWS CLI

인증서 옵션을 업데이트하려면

다음 update-certificate-options 명령은 인증서 투명성 로깅을 옵트아웃합니다.

```
aws acm update-certificate-options --certificate-
arn arn:aws:acm:region:account:certificate/12345678-1234-1234-1234-123456789012 --
options CertificateTransparencyLoggingPreference=DISABLED
```

- 자세한 API 내용은 명령 참조UpdateCertificateOptions의 섹션을 참조하세요. AWS CLI

## API 를 사용한 게이트웨이 예제 AWS CLI

다음 코드 예제에서는 API Gateway AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

**create-api-key**

다음 코드 예시에서는 create-api-key을 사용하는 방법을 보여 줍니다.

AWS CLI

기존 API 및 스테이지에 대해 활성화된 API 키를 생성하려면

명령:

```
aws apigateway create-api-key --name 'Dev API Key' --description 'Used for
 development' --enabled --stage-keys restApiId='a1b2c3d4e5',stageName='dev'
```

- 자세한 API 내용은 명령 참조CreateApiKey의 섹션을 참조하세요. AWS CLI

### create-authorizer

다음 코드 예시에서는 create-authorizer을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 에 대한 토큰 기반 API Gateway Custom Authorizer를 생성하려면 API

다음 create-authorizer 예제에서는 토큰 기반 권한 부여자를 생성합니다.

```
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
```

출력:

```
{
    "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"
```

```
}
```

예제 2: 에 대한 Cognito 사용자 풀 기반 API Gateway Custom Authorizer를 생성하려면 API

다음 `create-authorizer` 예제에서는 Cognito 사용자 풀 기반 API Gateway Custom Authorizer를 생성합니다.

```
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'
```

출력:

```
{
    "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"
}
```

예제 3: 에 대한 요청 기반 API Gateway Custom Authorizer를 생성하려면 API

다음 `create-authorizer` 예제에서는 요청 기반 권한 부여자를 생성합니다.

```
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
```

출력:

```
{
    "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
}
```

- 자세한 API 내용은 명령 참조CreateAuthorizer의 섹션을 참조하세요. AWS CLI

### `create-base-path-mapping`

다음 코드 예시에서는 `create-base-path-mapping`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 도메인 이름에 대한 기본 경로 매핑을 생성하려면

명령:

```
aws apigateway create-base-path-mapping --domain-name subdomain.domain.tld --rest-
api-id 1234123412 --stage prod --base-path v1
```

- 자세한 API 내용은 명령 참조CreateBasePathMapping의 섹션을 참조하세요. AWS CLI

### `create-deployment`

다음 코드 예시에서는 `create-deployment`을 사용하는 방법을 보여 줍니다.

AWS CLI

에 대해 구성된 리소스를 새 스테이지에 배포API하려면

명령:

```
aws apigateway create-deployment --rest-api-id 1234123412 --stage-name dev --stage-
description 'Development Stage' --description 'First deployment to the dev stage'
```

에 대해 구성된 리소스를 기존 스테이지에 배포API하려면

명령:

```
aws apigateway create-deployment --rest-api-id 1234123412 --stage-name dev --
description 'Second deployment to the dev stage'
```

스테이지 변수를 사용하여 에 대해 구성된 리소스를 기존 스테이지API에 배포하려면

aws apigateway create-deployment --rest-api-id 1234123412 --stage-name dev --description 'dev 단계에 대한 세 번째 배포' --variables key='value',otherKey=otherValue'

- 자세한 API 내용은 명령 참조CreateDeployment의 섹션을 참조하세요. AWS CLI

### create-domain-name

다음 코드 예시에서는 create-domain-name을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 도메인 이름을 생성하려면

명령:

```
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'
```

- 자세한 API 내용은 명령 참조CreateDomainName의 섹션을 참조하세요. AWS CLI

### create-model

다음 코드 예시에서는 create-model을 사용하는 방법을 보여 줍니다.

AWS CLI

에 대한 모델을 생성하려면 API

명령:

```
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" } } } } }'
```

출력:

```
{
    "contentType": "application/json",
    "description": "The First Model",
    "name": "firstModel",
    "id": "2rzg0l",
    "schema": "{ \"$schema\": \"http://json-schema.org/draft-04/schema#\", \"title
\": \"firstModel\", \"type\": \"object\", \"properties\": { \"firstProperty
\" : { \"type\": \"object\", \"properties\": { \"key\": { \"type\": \"string
\" } } } } }"
}
```

- 자세한 API 내용은 명령 참조CreateModel의 섹션을 참조하세요. AWS CLI

### `create-resource`

다음 코드 예시에서는 `create-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 리소스를 생성하려면 API

명령:

```
aws apigateway create-resource --rest-api-id 1234123412 --parent-id a1b2c3 --path-
part 'new-resource'
```

- 자세한 API 내용은 명령 참조CreateResource의 섹션을 참조하세요. AWS CLI

### `create-rest-api`

다음 코드 예시에서는 `create-rest-api`을 사용하는 방법을 보여 줍니다.

AWS CLI

를 생성하려면 API

명령:

```
aws apigateway create-rest-api --name 'My First API' --description 'This is my first
 API'
```

기존 API에서 중복을 생성하려면 API

명령:

```
aws apigateway create-rest-api --name 'Copy of My First API' --description 'This is
 a copy of my first API' --clone-from 1234123412
```

- 자세한 API 내용은 명령 참조CreateRestApi의 섹션을 참조하세요. AWS CLI

### create-stage

다음 코드 예시에서는 create-stage을 사용하는 방법을 보여 줍니다.

AWS CLI

기존 배포를 API 포함할 에서 스테이지를 생성하려면

명령:

```
aws apigateway create-stage --rest-api-id 1234123412 --stage-name 'dev' --
description 'Development stage' --deployment-id a1b2c3
```

기존 배포 및 사용자 지정 스테이지 변수를 API 포함하는 의 스테이지를 생성하려면

명령:

```
aws apigateway create-stage --rest-api-id 1234123412 --stage-name 'dev'
 --description 'Development stage' --deployment-id a1b2c3 --variables
 key='value',otherKey='otherValue'
```

- 자세한 API 내용은 명령 참조CreateStage의 섹션을 참조하세요. AWS CLI

### create-usage-plan-key

다음 코드 예시에서는 create-usage-plan-key을 사용하는 방법을 보여 줍니다.

AWS CLI

기존 API 키를 사용 계획과 연결

명령:

```
aws apigateway create-usage-plan-key --usage-plan-id a1b2c3 --key-type "API_KEY" --
key-id 4vq3yryqm5
```

- 자세한 API 내용은 명령 참조CreateUsagePlanKey의 섹션을 참조하세요. AWS CLI

### `create-usage-plan`

다음 코드 예시에서는 `create-usage-plan`을 사용하는 방법을 보여 줍니다.

AWS CLI

매월 초에 재설정되는 스로틀 및 할당량 제한으로 사용 계획을 생성하려면

명령:

```
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
```

- 자세한 API 내용은 명령 참조CreateUsagePlan의 섹션을 참조하세요. AWS CLI

### `delete-api-key`

다음 코드 예시에서는 `delete-api-key`을 사용하는 방법을 보여 줍니다.

AWS CLI

API 키를 삭제하려면

명령:

```
aws apigateway delete-api-key --api-key 8bklk8bl1k3sB38D9B3l0enyWT8c09B30lkq0blk
```

- 자세한 API 내용은 명령 참조DeleteApiKey의 섹션을 참조하세요. AWS CLI

### `delete-authorizer`

다음 코드 예시에서는 `delete-authorizer`을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 사용자 지정 권한 부여자를 삭제하려면 API

명령:

```
aws apigateway delete-authorizer --rest-api-id 1234123412 --authorizer-id 7gkfbo
```

- 자세한 API 내용은 명령 참조DeleteAuthorizer의 섹션을 참조하세요. AWS CLI

### `delete-base-path-mapping`

다음 코드 예시에서는 `delete-base-path-mapping`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 도메인 이름에 대한 기본 경로 매핑을 삭제하려면

명령:

```
aws apigateway delete-base-path-mapping --domain-name 'api.domain.tld' --base-path
 'dev'
```

- 자세한 API 내용은 명령 참조DeleteBasePathMapping의 섹션을 참조하세요. AWS CLI

### `delete-client-certificate`

다음 코드 예시에서는 `delete-client-certificate`을 사용하는 방법을 보여 줍니다.

AWS CLI

클라이언트 인증서를 삭제하려면

명령:

```
aws apigateway delete-client-certificate --client-certificate-id a1b2c3
```

- 자세한 API 내용은 명령 참조DeleteClientCertificate의 섹션을 참조하세요. AWS CLI

### delete-deployment

다음 코드 예시에서는 delete-deployment을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 배포를 삭제하려면 API

명령:

```
aws apigateway delete-deployment --rest-api-id 1234123412 --deployment-id a1b2c3
```

- 자세한 API 내용은 명령 참조DeleteDeployment의 섹션을 참조하세요. AWS CLI

### delete-domain-name

다음 코드 예시에서는 delete-domain-name을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 도메인 이름을 삭제하려면

명령:

```
aws apigateway delete-domain-name --domain-name 'api.domain.tld'
```

- 자세한 API 내용은 명령 참조DeleteDomainName의 섹션을 참조하세요. AWS CLI

### delete-integration-response

다음 코드 예시에서는 delete-integration-response을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 지정된 리소스, 메서드 및 상태 코드에 대한 통합 응답을 삭제하려면 API

명령:

```
aws apigateway delete-integration-response --rest-api-id 1234123412 --resource-
id a1b2c3 --http-method GET --status-code 200
```

- 자세한 API 내용은 명령 참조DeleteIntegrationResponse의 섹션을 참조하세요. AWS CLI

### delete-integration

다음 코드 예시에서는 delete-integration을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 지정된 리소스 및 메서드에 대한 통합을 삭제하려면 API

명령:

```
aws apigateway delete-integration --rest-api-id 1234123412 --resource-id a1b2c3 --
http-method GET
```

- 자세한 API 내용은 명령 참조DeleteIntegration의 섹션을 참조하세요. AWS CLI

### delete-method-response

다음 코드 예시에서는 delete-method-response을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 지정된 리소스, 메서드 및 상태 코드에 대한 메서드 응답을 삭제하려면 API

명령:

```
aws apigateway delete-method-response --rest-api-id 1234123412 --resource-id a1b2c3
 --http-method GET --status-code 200
```

- 자세한 API 내용은 명령 참조DeleteMethodResponse의 섹션을 참조하세요. AWS CLI

### delete-method

다음 코드 예시에서는 delete-method을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 지정된 리소스에 대한 메서드를 삭제하려면 API

명령:

```
aws apigateway delete-method --rest-api-id 1234123412 --resource-id a1b2c3 --http-
method GET
```

- 자세한 API 내용은 명령 참조DeleteMethod의 섹션을 참조하세요. AWS CLI

### `delete-model`

다음 코드 예시에서는 `delete-model`을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 에서 모델을 삭제하려면 API

명령:

```
aws apigateway delete-model --rest-api-id 1234123412 --model-name 'customModel'
```

- 자세한 API 내용은 명령 참조DeleteModel의 섹션을 참조하세요. AWS CLI

### `delete-resource`

다음 코드 예시에서는 `delete-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 리소스를 삭제하려면 API

명령:

```
aws apigateway delete-resource --rest-api-id 1234123412 --resource-id a1b2c3
```

- 자세한 API 내용은 명령 참조DeleteResource의 섹션을 참조하세요. AWS CLI

### `delete-rest-api`

다음 코드 예시에서는 `delete-rest-api`을 사용하는 방법을 보여 줍니다.

AWS CLI

를 삭제하려면 API

명령:

```
aws apigateway delete-rest-api --rest-api-id 1234123412
```

- 자세한 API 내용은 명령 참조DeleteRestApi의 섹션을 참조하세요. AWS CLI

### delete-stage

다음 코드 예시에서는 delete-stage을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 스테이지를 삭제하려면 API

명령:

```
aws apigateway delete-stage --rest-api-id 1234123412 --stage-name 'dev'
```

- 자세한 API 내용은 명령 참조DeleteStage의 섹션을 참조하세요. AWS CLI

### delete-usage-plan-key

다음 코드 예시에서는 delete-usage-plan-key을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 계획에서 API 키를 제거하려면

명령:

```
aws apigateway delete-usage-plan-key --usage-plan-id a1b2c3 --key-
id 1NbjQzMReAkeEQPNAW8r3dXsU2rDD7fc7f2Sipnu
```

- 자세한 API 내용은 명령 참조DeleteUsagePlanKey의 섹션을 참조하세요. AWS CLI

### delete-usage-plan

다음 코드 예시에서는 delete-usage-plan을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 계획을 삭제하려면

명령:

```
aws apigateway delete-usage-plan --usage-plan-id a1b2c3
```

- 자세한 API 내용은 명령 참조DeleteUsagePlan의 섹션을 참조하세요. AWS CLI

### `flush-stage-authorizers-cache`

다음 코드 예시에서는 `flush-stage-authorizers-cache`을 사용하는 방법을 보여 줍니다.

AWS CLI

스테이지의 모든 권한 부여자 캐시 항목을 플러시하려면

명령:

```
aws apigateway flush-stage-authorizers-cache --rest-api-id 1234123412 --stage-
name dev
```

- 자세한 API 내용은 명령 참조FlushStageAuthorizersCache의 섹션을 참조하세요. AWS CLI

### `flush-stage-cache`

다음 코드 예시에서는 `flush-stage-cache`을 사용하는 방법을 보여 줍니다.

AWS CLI

의 API단계에 대한 캐시를 플러시하려면

명령:

```
aws apigateway flush-stage-cache --rest-api-id 1234123412 --stage-name dev
```

- 자세한 API 내용은 명령 참조FlushStageCache의 섹션을 참조하세요. AWS CLI

### `generate-client-certificate`

다음 코드 예시에서는 `generate-client-certificate`을 사용하는 방법을 보여 줍니다.

AWS CLI

클라이언트 측 SSL 인증서를 생성하려면

명령:

```
aws apigateway generate-client-certificate --description 'My First Client
 Certificate'
```

- 자세한 API 내용은 명령 참조GenerateClientCertificate의 섹션을 참조하세요. AWS CLI

### get-account

다음 코드 예시에서는 get-account을 사용하는 방법을 보여 줍니다.

AWS CLI

API Gateway 계정 설정을 가져오려면

명령:

```
aws apigateway get-account
```

출력:

```
{
    "cloudwatchRoleArn": "arn:aws:iam::123412341234:role/
APIGatewayToCloudWatchLogsRole",
    "throttleSettings": {
        "rateLimit": 500.0,
        "burstLimit": 1000
    }
}
```

- 자세한 API 내용은 명령 참조GetAccount의 섹션을 참조하세요. AWS CLI

### get-api-key

다음 코드 예시에서는 get-api-key을 사용하는 방법을 보여 줍니다.

AWS CLI

특정 API 키에 대한 정보를 가져오려면

명령:

```
aws apigateway get-api-key --api-key 8bklk8bl1k3sB38D9B3l0enyWT8c09B30lkq0blk
```

출력:

```
{
    "description": "My first key",
    "enabled": true,
    "stageKeys": [
        "a1b2c3d4e5/dev",
        "e5d4c3b2a1/dev"
    ],
    "lastUpdatedDate": 1456184515,
    "createdDate": 1456184452,
    "id": "8bklk8bl1k3sB38D9B3l0enyWT8c09B30lkq0blk",
    "name": "My key"
}
```

- 자세한 API 내용은 명령 참조GetApiKey의 섹션을 참조하세요. AWS CLI

### get-api-keys

다음 코드 예시에서는 get-api-keys을 사용하는 방법을 보여 줍니다.

AWS CLI

API 키 목록을 가져오려면

명령:

```
aws apigateway get-api-keys
```

출력:

```
{
```

```
    "items": [
        {
            "description": "My first key",
            "enabled": true,
            "stageKeys": [
                "a1b2c3d4e5/dev",
                "e5d4c3b2a1/dev"
            ],
            "lastUpdatedDate": 1456184515,
            "createdDate": 1456184452,
            "id": "8bklk8bl1k3sB38D9B3l0enyWT8c09B30lkq0blk",
            "name": "My key"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetApiKeys의 섹션을 참조하세요. AWS CLI

### `get-authorizer`

다음 코드 예시에서는 `get-authorizer`을 사용하는 방법을 보여 줍니다.

AWS CLI

권한 부여자별 API API 게이트웨이 설정을 가져오려면

명령:

```
aws apigateway get-authorizer --rest-api-id 1234123412 --authorizer-id gfi4n3
```

출력:

```
{
    "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"
}
```

- 자세한 API 내용은 명령 참조GetAuthorizer의 섹션을 참조하세요. AWS CLI

### `get-authorizers`

다음 코드 예시에서는 `get-authorizers`을 사용하는 방법을 보여 줍니다.

AWS CLI

에 대한 권한 부여자 목록을 가져오려면 REST API

명령:

```
aws apigateway get-authorizers --rest-api-id 1234123412
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetAuthorizers의 섹션을 참조하세요. AWS CLI

### `get-base-path-mapping`

다음 코드 예시에서는 `get-base-path-mapping`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 도메인 이름의 기본 경로 매핑을 가져오는 방법

명령:

```
aws apigateway get-base-path-mapping --domain-name subdomain.domain.tld --base-
path v1
```

출력:

```
{
    "basePath": "v1",
    "restApiId": "1234w4321e",
    "stage": "api"
}
```

- 자세한 API 내용은 명령 참조GetBasePathMapping의 섹션을 참조하세요. AWS CLI

### get-base-path-mappings

다음 코드 예시에서는 get-base-path-mappings을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 도메인 이름에 대한 기본 경로 매핑을 가져오려면

명령:

```
aws apigateway get-base-path-mappings --domain-name subdomain.domain.tld
```

출력:

```
{
    "items": [
        {
            "basePath": "(none)",
            "restApiId": "1234w4321e",
            "stage": "dev"
        },
        {
            "basePath": "v1",
            "restApiId": "1234w4321e",
            "stage": "api"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetBasePathMappings의 섹션을 참조하세요. AWS CLI

### get-client-certificate

다음 코드 예시에서는 get-client-certificate을 사용하는 방법을 보여 줍니다.

AWS CLI

클라이언트 인증서를 가져오려면

명령:

```
aws apigateway get-client-certificate --client-certificate-id a1b2c3
```

- 자세한 API 내용은 명령 참조GetClientCertificate의 섹션을 참조하세요. AWS CLI

### get-client-certificates

다음 코드 예시에서는 get-client-certificates을 사용하는 방법을 보여 줍니다.

AWS CLI

클라이언트 인증서 목록을 가져오려면

명령:

```
aws apigateway get-client-certificates
```

출력:

```
{
    "items": [
        {
            "pemEncodedCertificate": "-----BEGIN CERTIFICATE----- <certificate
 content> -----END CERTIFICATE-----",
            "clientCertificateId": "a1b2c3",
            "expirationDate": 1483556561,
            "description": "My Client Certificate",
            "createdDate": 1452020561
        }
    ]
```

```
}
```

- 자세한 API 내용은 명령 참조GetClientCertificates의 섹션을 참조하세요. AWS CLI

### `get-deployment`

다음 코드 예시에서는 `get-deployment`을 사용하는 방법을 보여 줍니다.

AWS CLI

배포에 대한 정보를 가져오려면

명령:

```
aws apigateway get-deployment --rest-api-id 1234123412 --deployment-id ztt4m2
```

출력:

```
{
    "description": "myDeployment",
    "id": "ztt4m2",
    "createdDate": 1455218022
}
```

- 자세한 API 내용은 명령 참조GetDeployment의 섹션을 참조하세요. AWS CLI

### `get-deployments`

다음 코드 예시에서는 `get-deployments`을 사용하는 방법을 보여 줍니다.

AWS CLI

에 대한 배포 목록을 가져오려면 REST API

명령:

```
aws apigateway get-deployments --rest-api-id 1234123412
```

출력:

```
{
```

```
    "items": [
        {
            "createdDate": 1453797217,
            "id": "0a2b4c",
            "description": "Deployed my API for the first time"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetDeployments의 섹션을 참조하세요. AWS CLI

### get-domain-name

다음 코드 예시에서는 get-domain-name을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 도메인 이름에 대한 정보를 가져오려면

명령:

```
aws apigateway get-domain-name --domain-name api.domain.tld
```

출력:

```
{
    "domainName": "api.domain.tld",
    "distributionDomainName": "d1a2f3a4c5o6d.cloudfront.net",
    "certificateName": "uploadedCertificate",
    "certificateUploadDate": 1462565487
}
```

- 자세한 API 내용은 명령 참조GetDomainName의 섹션을 참조하세요. AWS CLI

### get-domain-names

다음 코드 예시에서는 get-domain-names을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 도메인 이름 목록을 가져오려면

명령:

```
aws apigateway get-domain-names
```

출력:

```
{
    "items": [
        {
            "distributionDomainName": "d9511k3l09bkd.cloudfront.net",
            "certificateUploadDate": 1452812505,
            "certificateName": "my_custom_domain-certificate",
            "domainName": "subdomain.domain.tld"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetDomainNames의 섹션을 참조하세요. AWS CLI

### get-export

다음 코드 예시에서는 get-export을 사용하는 방법을 보여 줍니다.

AWS CLI

단계에 대한 JSON 스웨거 템플릿을 가져오려면

명령:

```
aws apigateway get-export --rest-api-id a1b2c3d4e5 --stage-name dev --export-
type swagger /path/to/filename.json
```

스테이지의 JSON 스웨거 템플릿 + API 게이트웨이 확장을 가져오려면

명령:

```
aws apigateway get-export --parameters extensions='integrations' --rest-api-
id a1b2c3d4e5 --stage-name dev --export-type swagger /path/to/filename.json
```

단계에 대한 JSON 스웨거 템플릿 + Postman 익스텐션을 가져오려면

명령:

```
aws apigateway get-export --parameters extensions='postman' --rest-api-id a1b2c3d4e5
 --stage-name dev --export-type swagger /path/to/filename.json
```

- 자세한 API 내용은 명령 참조GetExport의 섹션을 참조하세요. AWS CLI

### get-integration-response

다음 코드 예시에서는 get-integration-response을 사용하는 방법을 보여 줍니다.

AWS CLI

의 REST API리소스에 정의된 HTTP 메서드에 대한 통합 응답 구성을 가져오려면

명령:

```
aws apigateway get-integration-response --rest-api-id 1234123412 --resource-
id y9h6rt --http-method GET --status-code 200
```

출력:

```
{
    "statusCode": "200",
    "responseTemplates": {
        "application/json": null
    }
}
```

- 자세한 API 내용은 명령 참조GetIntegrationResponse의 섹션을 참조하세요. AWS CLI

### get-integration

다음 코드 예시에서는 get-integration을 사용하는 방법을 보여 줍니다.

AWS CLI

의 리소스에 정의된 HTTP 메서RESTAPI드에 대한 통합 구성을 가져오려면

명령:

```
aws apigateway get-integration --rest-api-id 1234123412 --resource-id y9h6rt --http-
method GET
```

출력:

```
{
    "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"
}
```

- 자세한 API 내용은 명령 참조GetIntegration의 섹션을 참조하세요. AWS CLI

### get-method-response

다음 코드 예시에서는 get-method-response을 사용하는 방법을 보여 줍니다.

AWS CLI

의 리소스에 정의된 메서드에 대한 HTTP 메서RESTAPI드 응답 리소스 구성을 가져오려면

명령:

```
aws apigateway get-method-response --rest-api-id 1234123412 --resource-id y9h6rt --
http-method GET --status-code 200
```

출력:

```
{
    "responseModels": {
        "application/json": "Empty"
    },
    "statusCode": "200"
```

```
}
```

- 자세한 API 내용은 명령 참조GetMethodResponse의 섹션을 참조하세요. AWS CLI

### get-method

다음 코드 예시에서는 get-method을 사용하는 방법을 보여 줍니다.

AWS CLI

의 리소스에 정의된 메서드에 대한 HTTP 메서RESTAPI드 리소스 구성을 가져오려면

명령:

```
aws apigateway get-method --rest-api-id 1234123412 --resource-id y9h6rt --http-
method GET
```

출력:

```
{
    "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"
}
```

- 자세한 API 내용은 명령 참조GetMethod의 섹션을 참조하세요. AWS CLI

### get-model-template

다음 코드 예시에서는 get-model-template을 사용하는 방법을 보여 줍니다.

AWS CLI

에 정의된 모델의 매핑 템플릿을 가져오려면 REST API

명령:

```
aws apigateway get-model-template --rest-api-id 1234123412 --model-name Empty
```

출력:

```
{
    "value": "#set($inputRoot = $input.path('$'))\n{ }"
}
```

- 자세한 API 내용은 명령 참조GetModelTemplate의 섹션을 참조하세요. AWS CLI

### get-model

다음 코드 예시에서는 get-model을 사용하는 방법을 보여 줍니다.

AWS CLI

에 정의된 모델의 구성을 가져오려면 REST API

명령:

```
aws apigateway get-model --rest-api-id 1234123412 --model-name Empty
```

출력:

```
{
    "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}"
}
```

- 자세한 API 내용은 명령 참조GetModel의 섹션을 참조하세요. AWS CLI

### `get-models`

다음 코드 예시에서는 `get-models`을 사용하는 방법을 보여 줍니다.

AWS CLI

에 대한 모델 목록을 가져오려면 REST API

명령:

```
aws apigateway get-models --rest-api-id 1234123412
```

출력:

```
{
    "items": [
        {
            "description": "This is a default error schema model",
            "schema": "{\n  \"$schema\" : \"http://json-schema.org/draft-04/schema#
\",\n  \"title\" : \"Error Schema\",\n  \"type\" : \"object\",\n  \"properties\" :
 {\n    \"message\" : { \"type\" : \"string\" }\n  }\n}",
            "contentType": "application/json",
            "id": "7tpbze",
            "name": "Error"
        },
        {
            "description": "This is a default empty schema model",
            "schema": "{\n  \"$schema\": \"http://json-schema.org/draft-04/schema#
\",\n  \"title\" : \"Empty Schema\",\n  \"type\" : \"object\"\n}",
```

```
            "contentType": "application/json",
            "id": "etd5w5",
            "name": "Empty"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetModels의 섹션을 참조하세요. AWS CLI

### get-resource

다음 코드 예시에서는 get-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 대한 정보를 가져오려면

명령:

```
aws apigateway get-resource --rest-api-id 1234123412 --resource-id zwo0y3
```

출력:

```
{
    "path": "/path",
    "pathPart": "path",
    "id": "zwo0y3",
    "parentId": "uyokt6ij2g"
}
```

- 자세한 API 내용은 명령 참조GetResource의 섹션을 참조하세요. AWS CLI

### get-resources

다음 코드 예시에서는 get-resources을 사용하는 방법을 보여 줍니다.

AWS CLI

에 대한 리소스 목록을 가져오려면 REST API

명령:

```
aws apigateway get-resources --rest-api-id 1234123412
```

출력:

```
{
    "items": [
        {
            "path": "/resource/subresource",
            "resourceMethods": {
                "POST": {}
            },
            "id": "024ace",
            "pathPart": "subresource",
            "parentId": "ai5b02"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetResources의 섹션을 참조하세요. AWS CLI

### get-rest-api

다음 코드 예시에서는 get-rest-api을 사용하는 방법을 보여 줍니다.

AWS CLI

에 대한 정보를 가져오려면 API

명령:

```
aws apigateway get-rest-api --rest-api-id 1234123412
```

출력:

```
{
    "name": "myAPI",
    "id": "o1y243m4f5",
    "createdDate": 1453416433
}
```

- 자세한 API 내용은 명령 참조GetRestApi의 섹션을 참조하세요. AWS CLI

### get-rest-apis

다음 코드 예시에서는 get-rest-apis을 사용하는 방법을 보여 줍니다.

AWS CLI

의 목록을 가져오려면 REST APIs

명령:

```
aws apigateway get-rest-apis
```

출력:

```
{
    "items": [
        {
            "createdDate": 1438884790,
            "id": "12s44z21rb",
            "name": "My First API"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetRestApis의 섹션을 참조하세요. AWS CLI

### get-sdk

다음 코드 예시에서는 get-sdk을 사용하는 방법을 보여 줍니다.

AWS CLI

REST API 단계에 SDK 대한 Android를 가져오려면

명령:

```
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
```

출력:

```
{
    "contentType": "application/octet-stream",
    "contentDisposition": "attachment; filename=\"android_2016-02-22_23-52Z.zip\""
}
```

REST API 단계에 IOS SDK 대한 를 가져오려면

명령:

```
aws apigateway get-sdk --rest-api-id 1234123412 --stage-name dev --sdk-
type objectivec --parameters classPrefix='myprefix' /path/to/iOS_sdk.zip
```

출력:

```
{
    "contentType": "application/octet-stream",
    "contentDisposition": "attachment; filename=\"objectivec_2016-02-22_23-52Z.zip
\""
}
```

REST API 단계에 SDK 대한 Javascript를 가져오려면

명령:

```
aws apigateway get-sdk --rest-api-id 1234123412 --stage-name dev --sdk-
type javascript /path/to/javascript_sdk.zip
```

출력:

```
{
    "contentType": "application/octet-stream",
    "contentDisposition": "attachment; filename=\"javascript_2016-02-22_23-52Z.zip
\""
}
```

- 자세한 API 내용은 명령 참조GetSdk의 섹션을 참조하세요. AWS CLI

### `get-stage`

다음 코드 예시에서는 `get-stage`을 사용하는 방법을 보여 줍니다.

AWS CLI

API의 단계에 대한 정보를 가져오려면

명령:

```
aws apigateway get-stage --rest-api-id 1234123412 --stage-name dev
```

출력:

```
{
    "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
        }
    }
}
```

- 자세한 API 내용은 명령 참조GetStage의 섹션을 참조하세요. AWS CLI

### `get-stages`

다음 코드 예시에서는 `get-stages`을 사용하는 방법을 보여 줍니다.

AWS CLI

의 단계 목록을 가져오려면 REST API

명령:

```
aws apigateway get-stages --rest-api-id 1234123412
```

출력:

```
{
    "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
                }
```

```
            }
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetStages의 섹션을 참조하세요. AWS CLI

### get-usage-plan-key

다음 코드 예시에서는 get-usage-plan-key을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 계획과 연결된 API 키의 세부 정보를 가져오려면

명령:

```
aws apigateway get-usage-plan-key --usage-plan-id a1b2c3 --key-
id 1NbjQzMReAkeEQPNAW8r3dXsU2rDD7fc7f2Sipnu
```

- 자세한 API 내용은 명령 참조GetUsagePlanKey의 섹션을 참조하세요. AWS CLI

### get-usage-plan-keys

다음 코드 예시에서는 get-usage-plan-keys을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 계획과 연결된 API 키 목록을 가져오려면

명령:

```
aws apigateway get-usage-plan-keys --usage-plan-id a1b2c3
```

- 자세한 API 내용은 명령 참조GetUsagePlanKeys의 섹션을 참조하세요. AWS CLI

### get-usage-plan

다음 코드 예시에서는 get-usage-plan을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 계획의 세부 정보를 가져오려면

명령:

```
aws apigateway get-usage-plan --usage-plan-id a1b2c3
```

- 자세한 API 내용은 명령 참조GetUsagePlan의 섹션을 참조하세요. AWS CLI

### get-usage-plans

다음 코드 예시에서는 get-usage-plans을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 사용 계획의 세부 정보를 가져오려면

명령:

```
aws apigateway get-usage-plans
```

- 자세한 API 내용은 명령 참조GetUsagePlans의 섹션을 참조하세요. AWS CLI

### get-usage

다음 코드 예시에서는 get-usage을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 계획의 사용 세부 정보를 가져오려면

명령:

```
aws apigateway get-usage --usage-plan-id a1b2c3 --start-date "2016-08-16" --end-
date "2016-08-17"
```

- 자세한 API 내용은 명령 참조GetUsage의 섹션을 참조하세요. AWS CLI

### import-rest-api

다음 코드 예시에서는 import-rest-api을 사용하는 방법을 보여 줍니다.

AWS CLI

스웨거 템플릿을 가져오고 API

명령:

```
aws apigateway import-rest-api --body 'file:///path/to/API_Swagger_template.json'
```

- 자세한 API 내용은 명령 참조ImportRestApi의 섹션을 참조하세요. AWS CLI

### put-integration-response

다음 코드 예시에서는 put-integration-response을 사용하는 방법을 보여 줍니다.

AWS CLI

정의된 매핑 템플릿을 사용하여 통합 응답을 기본 응답으로 생성하는 방법

명령:

```
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\"}"}'
```

정규식이 400이고 헤더 값이 정적으로 정의된 통합 응답을 생성하는 방법

명령:

```
aws apigateway put-integration-response --rest-api-id 1234123412 --resource-
id a1b2c3 --http-method GET --status-code 400 --selection-pattern 400 --response-
parameters '{"method.response.header.custom-header": "'"'"'custom-value'"'"'"}'
```

- 자세한 API 내용은 명령 참조PutIntegrationResponse의 섹션을 참조하세요. AWS CLI

### put-integration

다음 코드 예시에서는 put-integration을 사용하는 방법을 보여 줍니다.

AWS CLI

MOCK 통합 요청을 생성하려면

명령:

```
aws apigateway put-integration --rest-api-id 1234123412 --resource-id a1b2c3 --http-
method GET --type MOCK --request-templates '{ "application/json": "{\"statusCode\":
 200}" }'
```

HTTP 통합 요청을 생성하려면

명령:

```
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'
```

Lambda 함수 엔드포인트와 AWS 통합 요청을 생성하려면

명령:

```
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'
```

- 자세한 API 내용은 명령 참조PutIntegration의 섹션을 참조하세요. AWS CLI

### put-method-response

다음 코드 예시에서는 put-method-response을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 메서드 응답 헤더를 사용하여 지정된 상태 코드에서 메서드 응답을 생성하는 방법

명령:

```
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"
```

- 자세한 API 내용은 명령 참조PutMethodResponse의 섹션을 참조하세요. AWS CLI

### put-method

다음 코드 예시에서는 put-method을 사용하는 방법을 보여 줍니다.

AWS CLI

권한 부여, API 키 및 사용자 지정 메서드 요청 헤더 API 없이 에서 리소스에 대한 메서드를 생성하려면

명령:

```
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"
```

- 자세한 API 내용은 명령 참조PutMethod의 섹션을 참조하세요. AWS CLI

### put-rest-api

다음 코드 예시에서는 put-rest-api을 사용하는 방법을 보여 줍니다.

AWS CLI

스웨거 템플릿을 API 사용하여 기존 을 덮어쓰려면

명령:

```
aws apigateway put-rest-api --rest-api-id 1234123412 --mode overwrite --body
 'fileb:///path/to/API_Swagger_template.json'
```

Swagger 템플릿을 기존 템플릿으로 병합하려면 API

명령:

```
aws apigateway put-rest-api --rest-api-id 1234123412 --mode merge --body 'fileb:///
path/to/API_Swagger_template.json'
```

- 자세한 API 내용은 명령 참조PutRestApi의 섹션을 참조하세요. AWS CLI

### test-invoke-authorizer

다음 코드 예시에서는 test-invoke-authorizer을 사용하는 방법을 보여 줍니다.

AWS CLI

필요한 헤더 및 값을 포함하여 사용자 지정 권한 부여자에게 요청 호출을 테스트하려면

명령:

```
aws apigateway test-invoke-authorizer --rest-api-id 1234123412 --authorizer-
id 5yid1t --headers Authorization='Value'
```

- 자세한 API 내용은 명령 참조TestInvokeAuthorizer의 섹션을 참조하세요. AWS CLI

### `test-invoke-method`

다음 코드 예시에서는 `test-invoke-method`을 사용하는 방법을 보여 줍니다.

AWS CLI

GET 요청을 API 수행하여 에서 루트 리소스 호출을 테스트하려면

명령:

```
aws apigateway test-invoke-method --rest-api-id 1234123412 --resource-id avl5sg8fw8
 --http-method GET --path-with-query-string '/'
```

경로 파라미터 값이 지정된 GET 요청을 API 수행하여 에서 하위 리소스 호출을 테스트하려면

명령:

```
aws apigateway test-invoke-method --rest-api-id 1234123412 --resource-id 3gapai --
http-method GET --path-with-query-string '/pets/1'
```

- 자세한 API 내용은 명령 참조TestInvokeMethod의 섹션을 참조하세요. AWS CLI

### `update-account`

다음 코드 예시에서는 `update-account`을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudWatch 로그에 로깅하기 ARN 위한 IAM 역할을 변경하려면

명령:

```
aws apigateway update-account --patch-operations op='replace',path='/
cloudwatchRoleArn',value='arn:aws:iam::123412341234:role/APIGatewayToCloudWatchLogs'
```

출력:

```
{
    "cloudwatchRoleArn": "arn:aws:iam::123412341234:role/
APIGatewayToCloudWatchLogs",
    "throttleSettings": {
        "rateLimit": 1000.0,
        "burstLimit": 2000
    }
}
```

- 자세한 API 내용은 명령 참조UpdateAccount의 섹션을 참조하세요. AWS CLI

### update-api-key

다음 코드 예시에서는 update-api-key을 사용하는 방법을 보여 줍니다.

AWS CLI

API 키의 이름을 변경하려면

명령:

```
aws apigateway update-api-key --api-key sNvjQDMReA1eEQPNAW8r37XsU2rDD7fc7m2SiMnu --
patch-operations op='replace',path='/name',value='newName'
```

출력:

```
{
    "description": "currentDescription",
    "enabled": true,
    "stageKeys": [
        "41t2j324r5/dev"
    ],
    "lastUpdatedDate": 1470086052,
    "createdDate": 1445460347,
    "id": "sNvjQDMReA1vEQPNzW8r3dXsU2rrD7fcjm2SiMnu",
    "name": "newName"
```

```
}
```

API 키를 비활성화하려면

명령:

```
aws apigateway update-api-key --api-key sNvjQDMReA1eEQPNAW8r37XsU2rDD7fc7m2SiMnu --
patch-operations op='replace',path='/enabled',value='false'
```

출력:

```
{
    "description": "currentDescription",
    "enabled": false,
    "stageKeys": [
        "41t2j324r5/dev"
    ],
    "lastUpdatedDate": 1470086052,
    "createdDate": 1445460347,
    "id": "sNvjQDMReA1vEQPNzW8r3dXsU2rrD7fcjm2SiMnu",
    "name": "newName"
}
```

- 자세한 API 내용은 명령 참조UpdateApiKey의 섹션을 참조하세요. AWS CLI

### update-authorizer

다음 코드 예시에서는 update-authorizer을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 권한 부여자의 이름을 변경하려면

명령:

```
aws apigateway update-authorizer --rest-api-id 1234123412 --authorizer-id gfi4n3 --
patch-operations op='replace',path='/name',value='testAuthorizer'
```

출력:

```
{
```

```
    "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"
}
```

사용자 지정 권한 부여자가 호출하는 Lambda 함수를 변경하려면

명령:

```
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'
```

출력:

```
{
    "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"
}
```

- 자세한 API 내용은 명령 참조UpdateAuthorizer의 섹션을 참조하세요. AWS CLI

### update-base-path-mapping

다음 코드 예시에서는 update-base-path-mapping을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 도메인 이름의 기본 경로를 변경하는 방법

명령:

```
aws apigateway update-base-path-mapping --domain-name api.domain.tld --base-
path prod --patch-operations op='replace',path='/basePath',value='v1'
```

출력:

```
{
    "basePath": "v1",
    "restApiId": "1234123412",
    "stage": "api"
}
```

- 자세한 API 내용은 명령 참조UpdateBasePathMapping의 섹션을 참조하세요. AWS CLI

### update-client-certificate

다음 코드 예시에서는 update-client-certificate을 사용하는 방법을 보여 줍니다.

AWS CLI

클라이언트 인증서에 대한 설명을 업데이트하려면

명령:

```
aws apigateway update-client-certificate --client-certificate-id a1b2c3 --patch-
operations op='replace',path='/description',value='My new description'
```

- 자세한 API 내용은 명령 참조UpdateClientCertificate의 섹션을 참조하세요. AWS CLI

### update-deployment

다음 코드 예시에서는 update-deployment을 사용하는 방법을 보여 줍니다.

AWS CLI

배포에 대한 설명을 변경하려면

명령:

```
aws apigateway update-deployment --rest-api-id 1234123412 --deployment-id ztt4m2 --
patch-operations op='replace',path='/description',value='newDescription'
```

출력:

```
{
    "description": "newDescription",
    "id": "ztt4m2",
    "createdDate": 1455218022
}
```

- 자세한 API 내용은 명령 참조UpdateDeployment의 섹션을 참조하세요. AWS CLI

### update-domain-name

다음 코드 예시에서는 update-domain-name을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 도메인 이름의 인증서 이름을 변경하려면

다음 update-domain-name 예제에서는 사용자 지정 도메인의 인증서 이름을 변경합니다.

```
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'
```

출력:

```
{
    "domainName": "api.domain.tld",
    "distributionDomainName": "d123456789012.cloudfront.net",
    "certificateArn": "arn:aws:acm:us-west-2:111122223333:certificate/
CERTEXAMPLE123EXAMPLE",
    "certificateUploadDate": 1462565487
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서의 GatewayAPI에서 에 대한 사용자 지정 도메인 이름 설정을 참조하세요. API

- 자세한 API 내용은 명령 참조UpdateDomainName의 섹션을 참조하세요. AWS CLI

### update-integration-response

다음 코드 예시에서는 update-integration-response을 사용하는 방법을 보여 줍니다.

AWS CLI

정적 매핑이 '*'가 되도록 통합 응답 헤더를 변경하려면

명령:

```
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='"'"'*'"'"'
```

출력:

```
{
    "statusCode": "200",
    "responseParameters": {
        "method.response.header.Access-Control-Allow-Origin": "'*'"
    }
}
```

통합 응답 헤더를 제거하려면

명령:

```
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'
```

- 자세한 API 내용은 명령 참조UpdateIntegrationResponse의 섹션을 참조하세요. AWS CLI

### update-integration

다음 코드 예시에서는 update-integration을 사용하는 방법을 보여 줍니다.

AWS CLI

Input Passthrough로 구성된 'Content-Type: application/json' 매핑 템플릿을 추가하려면

명령:

```
aws apigateway update-integration \
    --rest-api-id a1b2c3d4e5 \
    --resource-id a1b2c3 \
    --http-method POST \
    --patch-operations "op='add',path='/requestTemplates/application~1json'"
```

사용자 지정 템플릿으로 구성된 '콘텐츠 유형: 애플리케이션/json' 매핑 템플릿을 업데이트(교체)하려면

명령:

```
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"}'"
```

Input Passthrough로 'Content-Type: application/json'과 연결된 사용자 지정 템플릿을 업데이트(교체)하려면

명령:

```
aws apigateway update-integration \
    --rest-api-id a1b2c3d4e5 \
    --resource-id a1b2c3 \
    --http-method POST \
    --patch-operations "op='replace',path='requestTemplates/application~1json'"
```

'콘텐츠 유형: application/json' 매핑 템플릿을 제거하려면

명령:

```
aws apigateway update-integration \
    --rest-api-id a1b2c3d4e5 \
    --resource-id a1b2c3 \
    --http-method POST \
    --patch-operations "op='remove',path='/requestTemplates/application~1json'"
```

- 자세한 API 내용은 명령 참조UpdateIntegration의 섹션을 참조하세요. AWS CLI

### update-method-response

다음 코드 예시에서는 update-method-response을 사용하는 방법을 보여 줍니다.

AWS CLI

메서드에서 200 응답에 대한 새 메서드 응답 헤더를 생성하고 필요하지 않음으로 정의하려면(기본값)

명령:

```
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"
```

메서드에서 200 응답에 대한 응답 모델을 삭제하려면

명령:

```
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"
```

- 자세한 API 내용은 명령 참조UpdateMethodResponse의 섹션을 참조하세요. AWS CLI

### update-method

다음 코드 예시에서는 update-method을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: API 키를 요구하도록 메서드를 수정하려면

다음 update-method 예제에서는 API 키가 필요하도록 메서드를 수정합니다.

```
aws apigateway update-method \
    --rest-api-id 1234123412 \
    --resource-id a1b2c3 \
    --http-method GET \
    --patch-operations op="replace",path="/apiKeyRequired",value="true"
```

출력:

```
{
    "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": {}
            }
        }
    }
}
```

예제 2: IAM 인증이 필요한 메서드를 수정하려면

다음 update-method 예제에서는 IAM 인증이 필요한 메서드를 수정합니다.

```
aws apigateway update-method \
    --rest-api-id 1234123412 \
    --resource-id a1b2c3 \
    --http-method GET \
    --patch-operations op="replace",path="/authorizationType",value="AWS_IAM"
```

출력:

```
{
    "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": {}
            }
        }
    }
}
```

예제 3: Lambda 인증이 필요한 메서드를 수정하려면

다음 update-method 예제에서는 메서드를 필수 Lambda 권한 부여로 수정합니다.

```
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"
```

출력:

```
{
    "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": {}
            }
        }
    }
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서의 Gateway CLI REST 및 Gateway에서 에 대한 액세스 제어 및 관리를 사용하여 사용 계획 생성, 구성 및 테스트를 API 참조하세요. REST API API API

- 자세한 API 내용은 명령 참조UpdateMethod의 섹션을 참조하세요. AWS CLI

### update-model

다음 코드 예시에서는 update-model을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 모델에 대한 설명을 변경하려면 API

명령:

```
aws apigateway update-model --rest-api-id 1234123412 --model-name 'Empty' --patch-
operations op=replace,path=/description,value='New Description'
```

에서 모델의 스키마를 변경하려면 API

명령:

```
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\" }"'
```

- 자세한 API 내용은 명령 참조UpdateModel의 섹션을 참조하세요. AWS CLI

### update-resource

다음 코드 예시에서는 update-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스를 이동하여 의 다른 상위 리소스 아래에 배치하려면 API

명령:

```
aws apigateway update-resource --rest-api-id 1234123412 --resource-id 1a2b3c --
patch-operations op=replace,path=/parentId,value='3c2b1a'
```

출력:

```
{
    "path": "/resource",
    "pathPart": "resource",
    "id": "1a2b3c",
    "parentId": "3c2b1a"
}
```

에서 리소스(pathPart)의 이름을 바꾸려면 API

명령:

```
aws apigateway update-resource --rest-api-id 1234123412 --resource-id 1a2b3c --
patch-operations op=replace,path=/pathPart,value=newresourcename
```

출력:

```
{
    "path": "/newresourcename",
    "pathPart": "newresourcename",
    "id": "1a2b3c",
    "parentId": "3c2b1a"
}
```

- 자세한 API 내용은 명령 참조UpdateResource의 섹션을 참조하세요. AWS CLI

### update-rest-api

다음 코드 예시에서는 update-rest-api을 사용하는 방법을 보여 줍니다.

AWS CLI

의 이름을 변경하려면 API

명령:

```
aws apigateway update-rest-api --rest-api-id 1234123412 --patch-operations
 op=replace,path=/name,value='New Name'
```

에 대한 설명을 변경하려면 API

명령:

```
aws apigateway update-rest-api --rest-api-id 1234123412 --patch-operations
 op=replace,path=/description,value='New Description'
```

- 자세한 API 내용은 명령 참조UpdateRestApi의 섹션을 참조하세요. AWS CLI

### update-stage

다음 코드 예시에서는 update-stage을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 리소스 및 메서드의 스테이지 설정을 재정의하려면

다음 update-stage 예제는 스테이지 설정을 재정의하고 특정 리소스 및 메서드에 대한 전체 요청/응답 로깅을 끕니다.

```
aws apigateway update-stage \
    --rest-api-id 1234123412 \
    --stage-name 'dev' \
    --patch-operations op=replace,path=/~1resourceName/GET/logging/
dataTrace,value=false
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon Gateway 개발자 안내서REST의 에 대한 단계 설정을 API 참조하세요. API

예제 2: 단계의 모든 리소스 및 메서드에 대한 API 스테이지 설정을 업데이트하려면

다음 update-stage 예제에서는 API 단계의 모든 리소스 및 메서드에 대해 전체 요청/응답 로깅을 활성화합니다.

```
aws apigateway update-stage \
```

```
    --rest-api-id 1234123412 \
    --stage-name 'dev' \
    --patch-operations 'op=replace,path=/*/*/logging/dataTrace,value=true'
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon Gateway 개발자 안내서REST의 에 대한 단계 설정을 API 참조하세요.  API

- 자세한 API 내용은 명령 참조UpdateStage의 섹션을 참조하세요. AWS CLI

### update-usage-plan

다음 코드 예시에서는 update-usage-plan을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 계획에 정의된 기간을 변경하려면

명령:

```
aws apigateway update-usage-plan --usage-plan-id a1b2c3 --patch-operations
 op="replace",path="/quota/period",value="MONTH"
```

사용 계획에 정의된 할당량 제한을 변경하려면

명령:

```
aws apigateway update-usage-plan --usage-plan-id a1b2c3 --patch-operations
 op="replace",path="/quota/limit",value="500"
```

사용 계획에 정의된 스로틀 속도 제한을 변경하려면

명령:

```
aws apigateway update-usage-plan --usage-plan-id a1b2c3 --patch-operations
 op="replace",path="/throttle/rateLimit",value="10"
```

사용 계획에 정의된 스로틀 버스트 제한을 변경하려면

명령:

```
aws apigateway update-usage-plan --usage-plan-id a1b2c3 --patch-operations
 op="replace",path="/throttle/burstLimit",value="20"
```

- 자세한 API 내용은 명령 참조UpdateUsagePlan의 섹션을 참조하세요. AWS CLI

### update-usage

다음 코드 예시에서는 update-usage을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 계획에 정의된 현재 기간 동안 API 키의 할당량을 일시적으로 수정하려면

명령:

```
aws apigateway update-usage --usage-plan-id a1b2c3 --key-
id 1NbjQzMReAkeEQPNAW8r3dXsU2rDD7fc7f2Sipnu --patch-operations op="replace",path="/
remaining",value="50"
```

- 자세한 API 내용은 명령 참조UpdateUsage의 섹션을 참조하세요. AWS CLI

## API 를 사용한 게이트웨이 HTTP 및 WebSocket API 예제 AWS CLI

다음 코드 예제에서는 API Gateway HTTP 및 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 WebSocket API.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `create-api-mapping`

다음 코드 예시에서는 `create-api-mapping`을 사용하는 방법을 보여 줍니다.

AWS CLI

에 대한 API 매핑을 생성하려면 API

다음 `create-api-mapping` 예제에서는 의 `test` 단계를 `regional.example.com` 사용자 지정 도메인 이름의 `/myApi` 경로API에 매핑합니다.

```
aws apigatewayv2 create-api-mapping \
    --domain-name regional.example.com \
    --api-mapping-key myApi \
    --api-id a1b2c3d4 \
    --stage test
```

출력:

```
{
```

```
    "ApiId": "a1b2c3d4",
    "ApiMappingId": "0qzs2sy7bh",
    "ApiMappingKey": "myApi"
    "Stage": "test"
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서의 Gateway에서 리전 사용자 지정 도메인 이름 설정을 참조하세요. API

- 자세한 API 내용은 명령 참조CreateApiMapping의 섹션을 참조하세요. AWS CLI

### create-api

다음 코드 예시에서는 create-api을 사용하는 방법을 보여 줍니다.

AWS CLI

를 생성하려면 HTTP API

다음 create-api 예제에서는 빠른 생성을 사용하여 HTTPAPI를 생성합니다. 빠른 생성을 사용하여 AWS Lambda 또는 HTTP 통합, 기본 캐치-올 라우팅 및 변경 사항을 자동으로 배포하도록 구성된 기본 단계를 API 사용하여 를 생성할 수 있습니다. 다음 명령은 빠른 생성을 사용하여 Lambda 함수와 통합HTTPAPI되는 를 생성합니다.

```
aws apigatewayv2 create-api \
    --name my-http-api \
    --protocol-type HTTP \
    --target arn:aws:lambda:us-west-2:123456789012:function:my-lambda-function
```

출력:

```
{
    "ApiEndpoint": "https://a1b2c3d4.execute-api.us-west-2.amazonaws.com",
    "ApiId": "a1b2c3d4",
    "ApiKeySelectionExpression": "$request.header.x-api-key",
    "CreatedDate": "2020-04-08T19:05:45+00:00",
    "Name": "my-http-api",
    "ProtocolType": "HTTP",
    "RouteSelectionExpression": "$request.method $request.path"
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서 의 GatewayHTTPAPI에서 의 개발을 참조하세요. API

를 생성하려면 WebSocket API

다음 `create-api` 예제에서는 지정된 이름으로 WebSocket API를 생성합니다.

```
aws apigatewayv2 create-api \
    --name "myWebSocketApi" \
    --protocol-type WEBSOCKET \
    --route-selection-expression '$request.body.action'
```

출력:

```
{
    "ApiKeySelectionExpression": "$request.header.x-api-key",
    "Name": "myWebSocketApi",
    "CreatedDate": "2018-11-15T06:23:51Z",
    "ProtocolType": "WEBSOCKET",
    "RouteSelectionExpression": "'$request.body.action'",
    "ApiId": "aabbccddee"
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서의 Gateway에서 생성 WebSocket API 섹션을 참조하세요. API

- 자세한 API 내용은 명령 참조CreateApi의 섹션을 참조하세요. AWS CLI

### `create-authorizer`

다음 코드 예시에서는 `create-authorizer`을 사용하는 방법을 보여 줍니다.

AWS CLI

에 대한 JWT 권한 부여자를 생성하려면 HTTP API

다음 `create-authorizer` 예제에서는 Amazon Cognito를 자격 증명 공급자로 사용하는 JWT 권한 부여자를 생성합니다.

```
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
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서의 JWT 권한 부여자를 HTTP APIs 사용하여에 대한 액세스 제어를 참조하세요.

- 자세한 API 내용은 명령 참조CreateAuthorizer의 섹션을 참조하세요. AWS CLI

### `create-deployment`

다음 코드 예시에서는 `create-deployment`을 사용하는 방법을 보여 줍니다.

AWS CLI

에 대한 배포를 생성하려면 API

다음 `create-deployment` 예제에서는 API 에 대한 배포를 생성하고 배포를 의 dev 단계와 연결합니다API.

```
aws apigatewayv2 create-deployment \
```

```
    --api-id a1b2c3d4 \
    --stage-name dev
```

출력:

```
{
    "AutoDeployed": false,
    "CreatedDate": "2020-04-06T23:38:08Z",
    "DeploymentId": "53lz9l",
    "DeploymentStatus": "DEPLOYED"
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서의 API 배포를 참조하세요.

- 자세한 API 내용은 명령 참조CreateDeployment의 섹션을 참조하세요. AWS CLI

### create-domain-name

다음 코드 예시에서는 create-domain-name을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 도메인 이름을 생성하려면

다음 create-domain-name 예제에서는 에 대한 리전 사용자 지정 도메인 이름을 생성합니다API.

```
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
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서의 Gateway에서 리전 사용자 지정 도메인 이름 설정을 참조하세요. API

- 자세한 API 내용은 명령 참조CreateDomainName의 섹션을 참조하세요. AWS CLI

### `create-integration`

다음 코드 예시에서는 `create-integration`을 사용하는 방법을 보여 줍니다.

AWS CLI

통합을 WebSocket API 생성하려면

다음 `create-integration` 예제에서는 에 대한 모의 통합을 생성합니다 WebSocket API.

```
aws apigatewayv2 create-integration \
    --api-id aabbccddee \
    --passthrough-behavior WHEN_NO_MATCH \
    --timeout-in-millis 29000 \
    --connection-type INTERNET \
    --integration-type MOCK
```

출력:

```
{
    "ConnectionType": "INTERNET",
    "IntegrationId": "0abcdef",
    "IntegrationResponseSelectionExpression": "${integration.response.statuscode}",
    "IntegrationType": "MOCK",
    "PassthroughBehavior": "WHEN_NO_MATCH",
    "PayloadFormatVersion": "1.0",
    "TimeoutInMillis": 29000
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서의 Gateway에서 통합 요청 설정을 WebSocket API 참조하세요. API

HTTP API 통합을 생성하려면

다음 create-integration 예제에서는 HTTP 에 대한 AWS Lambda 통합을 생성합니다API.

```
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
```

출력:

```
{
    "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
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서HTTP의 에 대한 통합 구성을 APIs 참조하세요.

- 자세한 API 내용은 명령 참조CreateIntegration의 섹션을 참조하세요. AWS CLI

### create-route

다음 코드 예시에서는 create-route을 사용하는 방법을 보여 줍니다.

AWS CLI

WebSocket 또는 에 대한 $기본 라우팅을 생성하려면 HTTP API

다음 create-route 예제에서는 WebSocket 또는 HTTP 에 대한 $default 경로를 생성합니다 API.

```
aws apigatewayv2 create-route \
```

```
    --api-id aabbccddee \
    --route-key '$default'
```

출력:

```
{
    "ApiKeyRequired": false,
    "AuthorizationType": "NONE",
    "RouteKey": "$default",
    "RouteId": "1122334"
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서의 에 대한 경로 작업을 WebSocket APIs 참조하세요.

에 대한 라우팅을 생성하려면 HTTP API

다음 `create-route` 예제에서는 POST 요청을 수락`signup`하는 라는 라우팅을 생성합니다.

```
aws apigatewayv2 create-route \
    --api-id aabbccddee \
    --route-key 'POST /signup'
```

출력:

```
{
    "ApiKeyRequired": false,
    "AuthorizationType": "NONE",
    "RouteKey": "POST /signup",
    "RouteId": "1122334"
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서HTTP의 에 대한 라우팅 작업을 APIs 참조하세요.

- 자세한 API 내용은 명령 참조CreateRoute의 섹션을 참조하세요. AWS CLI

### `create-stage`

다음 코드 예시에서는 `create-stage`을 사용하는 방법을 보여 줍니다.

AWS CLI

스테이지를 생성하려면

다음 create-stage 예제에서는 에 대한 dev라는 스테이지를 생성합니다API.

```
aws apigatewayv2 create-stage \
    --api-id a1b2c3d4 \
    --stage-name dev
```

출력:

```
{
    "CreatedDate": "2020-04-06T23:23:46Z",
    "DefaultRouteSettings": {
        "DetailedMetricsEnabled": false
    },
    "LastUpdatedDate": "2020-04-06T23:23:46Z",
    "RouteSettings": {},
    "StageName": "dev",
    "StageVariables": {},
    "Tags": {}
}
```

자세한 내용은 Amazon Gateway 개발자 안내서HTTP의 에 대한 단계 작업을 APIs 참조하세요. API

- 자세한 API 내용은 명령 참조CreateStage의 섹션을 참조하세요. AWS CLI

### create-vpc-link

다음 코드 예시에서는 create-vpc-link을 사용하는 방법을 보여 줍니다.

AWS CLI

에 대한 VPC 링크를 생성하려면 HTTP API

다음 create-vpc-link 예제에서는 HTTP 에 대한 VPC 링크를 생성합니다APIs.

```
aws apigatewayv2 create-vpc-link \
    --name MyVpcLink \
```

```
    --subnet-ids subnet-aaaa subnet-bbbb \
    --security-group-ids sg1234 sg5678
```

출력:

```
{
    "CreatedDate": "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"
}
```

자세한 내용은 Amazon Gateway 개발자 안내서HTTP의 에 대한 VPC 링크 작업을 APIs 참조하세요. API

- 자세한 API 내용은 명령 참조CreateVpcLink의 섹션을 참조하세요. AWS CLI

### delete-access-log-settings

다음 코드 예시에서는 delete-access-log-settings을 사용하는 방법을 보여 줍니다.

AWS CLI

에 대한 액세스 로깅을 비활성화하려면 API

다음 delete-access-log-settings 예제에서는 의 $default 단계에 대한 액세스 로그 설정을 삭제합니다API. 단계에 대한 액세스 로깅을 비활성화하려면 해당 액세스 로그 설정을 삭제합니다.

```
aws apigatewayv2 delete-access-log-settings \
    --api-id a1b2c3d4 \
```

```
    --stage-name '$default'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Gateway 개발자 안내서HTTP의 에 대한 로깅 구성을 API 참조하세요. API

- 자세한 API 내용은 명령 참조DeleteAccessLogSettings의 섹션을 참조하세요. AWS CLI

### `delete-api-mapping`

다음 코드 예시에서는 `delete-api-mapping`을 사용하는 방법을 보여 줍니다.

AWS CLI

API 매핑을 삭제하려면

다음 `delete-api-mapping` 예제에서는 `api.example.com` 사용자 지정 도메인 이름에 대한 API 매핑을 삭제합니다.

```
aws apigatewayv2 delete-api-mapping \
    --api-mapping-id a1b2c3 \
    --domain-name api.example.com
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon API Gateway 개발자 안내서의 Gateway에서 리전 사용자 지정 도메인 이름 설정을 참조하세요. API

- 자세한 API 내용은 명령 참조DeleteApiMapping의 섹션을 참조하세요. AWS CLI

### `delete-api`

다음 코드 예시에서는 `delete-api`을 사용하는 방법을 보여 줍니다.

AWS CLI

를 삭제하려면 API

다음 `delete-api` 예제에서는 를 삭제합니다API.

```
aws apigatewayv2 delete-api \
```

```
    --api-id a1b2c3d4
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon API Gateway 개발자 안내서의 작업 HTTP APIs 및 작업을 WebSocket APIs 참조하세요.

- 자세한 API 내용은 명령 참조DeleteApi의 섹션을 참조하세요. AWS CLI

### delete-authorizer

다음 코드 예시에서는 delete-authorizer을 사용하는 방법을 보여 줍니다.

AWS CLI

권한 부여자를 삭제하려면

다음 delete-authorizer 예제에서는 권한 부여자를 삭제합니다.

```
aws apigatewayv2 delete-authorizer \
    --api-id a1b2c3d4 \
    --authorizer-id a1b2c3
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon API Gateway 개발자 안내서의 JWT 권한 부여자를 HTTP APIs 사용하여에 대한 액세스 제어를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteAuthorizer의 섹션을 참조하세요. AWS CLI

### delete-cors-configuration

다음 코드 예시에서는 delete-cors-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

에 대한 CORS 구성을 삭제하려면 HTTP API

다음 delete-cors-configuration 예제에서는 CORS 구성을 삭제HTTPAPI하여 에 CORS 대해 를 비활성화합니다.

```
aws apigatewayv2 delete-cors-configuration \
```

```
    --api-id a1b2c3d4
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Gateway 개발자 안내서HTTP의 에 CORS 대한 구성을 API 참조하세요. API

- 자세한 API 내용은 명령 참조DeleteCorsConfiguration의 섹션을 참조하세요. AWS CLI

### delete-deployment

다음 코드 예시에서는 delete-deployment을 사용하는 방법을 보여 줍니다.

AWS CLI

배포를 삭제하려면

다음 delete-deployment 예제에서는 의 배포를 삭제합니다API.

```
aws apigatewayv2 delete-deployment \
    --api-id a1b2c3d4 \
    --deployment-id a1b2c3
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon API Gateway 개발자 안내서의 API 배포를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteDeployment의 섹션을 참조하세요. AWS CLI

### delete-domain-name

다음 코드 예시에서는 delete-domain-name을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 도메인 이름을 삭제하려면

다음 delete-domain-name 예제에서는 사용자 지정 도메인 이름을 삭제합니다.

```
aws apigatewayv2 delete-domain-name \
    --domain-name api.example.com
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon API Gateway 개발자 안내서의 Gateway에서 리전 사용자 지정 도메인 이름 설정을 참조하세요. API

- 자세한 API 내용은 명령 참조DeleteDomainName의 섹션을 참조하세요. AWS CLI

### delete-integration

다음 코드 예시에서는 delete-integration을 사용하는 방법을 보여 줍니다.

AWS CLI

통합을 삭제하려면

다음 delete-integration 예제에서는 API 통합을 삭제합니다.

```
aws apigatewayv2 delete-integration \
    --api-id a1b2c3d4 \
    --integration-id a1b2c3
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon API Gateway 개발자 안내서의 에 대한 통합 구성 HTTP APIs 및 통합 설정을 WebSocket API 참조하세요.

- 자세한 API 내용은 명령 참조DeleteIntegration의 섹션을 참조하세요. AWS CLI

### delete-route-settings

다음 코드 예시에서는 delete-route-settings을 사용하는 방법을 보여 줍니다.

AWS CLI

라우팅 설정을 삭제하려면

다음 delete-route-settings 예제에서는 지정된 경로에 대한 경로 설정을 삭제합니다.

```
aws apigatewayv2 delete-route-settings \
    --api-id a1b2c3d4 \
    --stage-name dev \
```

```
    --route-key 'GET /pets'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Gateway 개발자 안내서HTTP의 에 대한 경로 작업을 APIs 참조하세요. API

- 자세한 API 내용은 명령 참조DeleteRouteSettings의 섹션을 참조하세요. AWS CLI

### `delete-route`

다음 코드 예시에서는 `delete-route`을 사용하는 방법을 보여 줍니다.

AWS CLI

경로를 삭제하려면

다음 `delete-route` 예제에서는 API 라우팅을 삭제합니다.

```
aws apigatewayv2 delete-route \
    --api-id a1b2c3d4 \
    --route-id a1b2c3
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Gateway 개발자 안내서HTTP의 에 대한 경로 작업을 APIs 참조하세요. API

- 자세한 API 내용은 명령 참조DeleteRoute의 섹션을 참조하세요. AWS CLI

### `delete-stage`

다음 코드 예시에서는 `delete-stage`을 사용하는 방법을 보여 줍니다.

AWS CLI

스테이지를 삭제하려면

다음 `delete-stage` 예제에서는 의 `test` 단계를 삭제합니다API.

```
aws apigatewayv2 delete-stage \
```

```
    --api-id a1b2c3d4 \
    --stage-name test
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Gateway 개발자 안내서HTTP의 에 대한 단계 작업을 APIs 참조하세요. API

- 자세한 API 내용은 명령 참조DeleteStage의 섹션을 참조하세요. AWS CLI

### `delete-vpc-link`

다음 코드 예시에서는 `delete-vpc-link`을 사용하는 방법을 보여 줍니다.

AWS CLI

에 대한 VPC 링크를 삭제하려면 HTTP API

다음 `delete-vpc-link` 예제에서는 VPC 링크를 삭제합니다.

```
aws apigatewayv2 delete-vpc-link \
    --vpc-link-id abcd123
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Gateway 개발자 안내서HTTP의 에 대한 VPC 링크 작업을 APIs 참조하세요. API

- 자세한 API 내용은 명령 참조DeleteVpcLink의 섹션을 참조하세요. AWS CLI

### `export-api`

다음 코드 예시에서는 `export-api`을 사용하는 방법을 보여 줍니다.

AWS CLI

의 OpenAPI 정의를 내보내려면 HTTP API

다음 `export-api` 예제에서는 이라는 이름의 API 스테이지에 대한 OpenAPI 3.0 정의를 라는 이름의 `prod` YAML 파일로 내보냅니다`stage-definition.yaml`. 내보낸 정의 파일에는 기본적으로 API Gateway 확장이 포함됩니다.

```
aws apigatewayv2 export-api \
    --api-id a1b2c3d4 \
    --output-type YAML \
    --specification OAS30 \
    --stage-name prod \
    stage-definition.yaml
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon API Gateway 개발자 안내서의 게이트웨이HTTPAPI에서 내보내기를 참조하세요. API

- 자세한 API 내용은 명령 참조ExportApi의 섹션을 참조하세요. AWS CLI

### get-api-mapping

다음 코드 예시에서는 get-api-mapping을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 도메인 이름의 API 매핑에 대한 정보를 가져오려면

다음 get-api-mapping 예제에서는 api.example.com 사용자 지정 도메인 이름의 API 매핑에 대한 정보를 보여줍니다.

```
aws apigatewayv2 get-api-mapping \
    --api-mapping-id a1b2c3 \
    --domain-name api.example.com
```

출력:

```
{
    "ApiId": "a1b2c3d4",
    "ApiMappingId": "a1b2c3d5",
    "ApiMappingKey": "myTestApi"
    "Stage": "test"
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서의 Gateway에서 리전 사용자 지정 도메인 이름 설정을 참조하세요. API

- 자세한 API 내용은 명령 참조GetApiMapping의 섹션을 참조하세요. AWS CLI

### get-api-mappings

다음 코드 예시에서는 get-api-mappings을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 도메인 이름에 대한 API 매핑을 가져오려면

다음 get-api-mappings 예제에서는 api.example.com 사용자 지정 도메인 이름에 대한 모든 API 매핑 목록을 표시합니다.

```
aws apigatewayv2 get-api-mappings \
    --domain-name api.example.com
```

출력:

```
{
    "Items": [
        {
            "ApiId": "a1b2c3d4",
            "ApiMappingId": "a1b2c3d5",
            "ApiMappingKey": "myTestApi"
            "Stage": "test"
        },
        {
            "ApiId": "a5b6c7d8",
            "ApiMappingId": "a1b2c3d6",
            "ApiMappingKey": "myDevApi"
            "Stage": "dev"
        },
    ]
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서의 Gateway에서 리전 사용자 지정 도메인 이름 설정을 참조하세요. API

- 자세한 API 내용은 명령 참조GetApiMappings의 섹션을 참조하세요. AWS CLI

### get-api

다음 코드 예시에서는 get-api을 사용하는 방법을 보여 줍니다.

AWS CLI

에 대한 정보를 검색하려면 API

다음 get-api 예제에서는 에 대한 정보를 보여줍니다API.

```
aws apigatewayv2 get-api \
    --api-id a1b2c3d4
```

출력:

```
{
    "ApiEndpoint": "https://a1b2c3d4.execute-api.us-west-2.amazonaws.com",
    "ApiId": "a1b2c3d4",
    "ApiKeySelectionExpression": "$request.header.x-api-key",
    "CreatedDate": "2020-03-28T00:32:37Z",
    "Name": "my-api",
    "ProtocolType": "HTTP",
    "RouteSelectionExpression": "$request.method $request.path",
    "Tags": {
        "department": "finance"
    }
}
```

- 자세한 API 내용은 명령 참조GetApi의 섹션을 참조하세요. AWS CLI

### get-apis

다음 코드 예시에서는 get-apis을 사용하는 방법을 보여 줍니다.

AWS CLI

목록을 검색하려면 APIs

다음 get-apis 예제에서는 현재 사용자의 APIs를 모두 나열합니다.

```
aws apigatewayv2 get-apis
```

출력:

```
{
```

```
    "Items": [
        {
            "ApiEndpoint": "wss://a1b2c3d4.execute-api.us-west-2.amazonaws.com",
            "ApiId": "a1b2c3d4",
            "ApiKeySelectionExpression": "$request.header.x-api-key",
            "CreatedDate": "2020-04-07T20:21:59Z",
            "Name": "my-websocket-api",
            "ProtocolType": "WEBSOCKET",
            "RouteSelectionExpression": "$request.body.message",
            "Tags": {}
        },
        {
            "ApiEndpoint": "https://a1b2c3d5.execute-api.us-west-2.amazonaws.com",
            "ApiId": "a1b2c3d5",
            "ApiKeySelectionExpression": "$request.header.x-api-key",
            "CreatedDate": "2020-04-07T20:23:50Z",
            "Name": "my-http-api",
            "ProtocolType": "HTTP",
            "RouteSelectionExpression": "$request.method $request.path",
            "Tags": {}
        }
    ]
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서의 작업 HTTP APIs 및 작업을 WebSocket APIs 참조하세요.

- 자세한 API 내용은 명령 참조GetApis의 섹션을 참조하세요. AWS CLI

### get-authorizer

다음 코드 예시에서는 get-authorizer을 사용하는 방법을 보여 줍니다.

AWS CLI

권한 부여자에 대한 정보를 검색하려면

다음 get-authorizer 예제에서는 권한 부여자에 대한 정보를 보여줍니다.

```
aws apigatewayv2 get-authorizer \
    --api-id a1b2c3d4 \
    --authorizer-id a1b2c3
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서의 JWT 권한 부여자를 HTTP APIs 사용하여에 대한 액세스 제어를 참조하세요.

- 자세한 API 내용은 명령 참조GetAuthorizer의 섹션을 참조하세요. AWS CLI

### `get-authorizers`

다음 코드 예시에서는 `get-authorizers`을 사용하는 방법을 보여 줍니다.

AWS CLI

에 대한 권한 부여자 목록을 검색하려면 API

다음 `get-authorizers` 예제에서는 의 모든 권한 부여자 목록을 표시합니다API.

```
aws apigatewayv2 get-authorizers \
    --api-id a1b2c3d4
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서의 JWT 권한 부여자를 HTTP APIs 사용하여에 대한 액세스 제어를 참조하세요.

- 자세한 API 내용은 명령 참조GetAuthorizers의 섹션을 참조하세요. AWS CLI

### get-deployment

다음 코드 예시에서는 get-deployment을 사용하는 방법을 보여 줍니다.

AWS CLI

배포에 대한 정보를 검색하려면

다음 get-deployment 예제에서는 배포에 대한 정보를 보여줍니다.

```
aws apigatewayv2 get-deployment \
    --api-id a1b2c3d4 \
    --deployment-id abcdef
```

출력:

```
{
    "AutoDeployed": true,
    "CreatedDate": "2020-04-07T23:58:40Z",
    "DeploymentId": "abcdef",
    "DeploymentStatus": "DEPLOYED",
    "Description": "Automatic deployment triggered by changes to the Api
 configuration"
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서의 API 배포를 참조하세요.

- 자세한 API 내용은 명령 참조GetDeployment의 섹션을 참조하세요. AWS CLI

### get-deployments

다음 코드 예시에서는 get-deployments을 사용하는 방법을 보여 줍니다.

AWS CLI

배포 목록을 검색하려면

다음 get-deployments 예제에서는 의 모든 API배포 목록을 표시합니다.

```
aws apigatewayv2 get-deployments \
    --api-id a1b2c3d4
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서의 API 배포를 참조하세요.

- 자세한 API 내용은 명령 참조GetDeployments의 섹션을 참조하세요. AWS CLI

### get-domain-name

다음 코드 예시에서는 get-domain-name을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 도메인 이름에 대한 정보를 검색하려면

다음 get-domain-name 예제에서는 사용자 지정 도메인 이름에 대한 정보를 보여줍니다.

```
aws apigatewayv2 get-domain-name \
    --domain-name api.example.com
```

출력:

```
{
    "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": {}
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서의 Gateway에서 리전 사용자 지정 도메인 이름 설정을 참조하세요. API

- 자세한 API 내용은 명령 참조GetDomainName의 섹션을 참조하세요. AWS CLI

### get-domain-names

다음 코드 예시에서는 get-domain-names을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 도메인 이름 목록을 검색하려면

다음 get-domain-names 예제에서는 현재 사용자의 모든 사용자 지정 도메인 이름 목록을 표시합니다.

```
aws apigatewayv2 get-domain-names
```

출력:

```
{
    "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"
                }
            ]
        }
    ]
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서의 Gateway에서 리전 사용자 지정 도메인 이름 설정을 참조하세요. API

- 자세한 API 내용은 명령 참조GetDomainNames의 섹션을 참조하세요. AWS CLI

### get-integration

다음 코드 예시에서는 get-integration을 사용하는 방법을 보여 줍니다.

AWS CLI

통합에 대한 정보를 검색하려면

다음 get-integration 예제에서는 통합에 대한 정보를 보여줍니다.

```
aws apigatewayv2 get-integration \
    --api-id a1b2c3d4 \
    --integration-id a1b2c3
```

출력:

```
{
    "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
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서의 에 대한 통합 구성 HTTP APIs 및 통합 설정을 WebSocket API 참조하세요.

- 자세한 API 내용은 명령 참조GetIntegration의 섹션을 참조하세요. AWS CLI

### get-integrations

다음 코드 예시에서는 get-integrations을 사용하는 방법을 보여 줍니다.

AWS CLI

통합 목록을 검색하려면

다음 get-integrations 예제에서는 의 모든 API통합 목록을 표시합니다.

```
aws apigatewayv2 get-integrations \
    --api-id a1b2c3d4
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서의 에 대한 통합 구성 HTTP APIs 및 통합 설정을 WebSocket API 참조하세요.

- 자세한 API 내용은 명령 참조GetIntegrations의 섹션을 참조하세요. AWS CLI

### get-route

다음 코드 예시에서는 get-route을 사용하는 방법을 보여 줍니다.

AWS CLI

경로에 대한 정보를 검색하려면

다음 get-route 예제에서는 라우팅에 대한 정보를 보여줍니다.

```
aws apigatewayv2 get-route \
    --api-id a1b2c3d4 \
    --route-id 72jz1wk
```

출력:

```
{
    "ApiKeyRequired": false,
    "AuthorizationType": "NONE",
    "RouteId": "72jz1wk",
```

```
    "RouteKey": "ANY /pets",
    "Target": "integrations/a1b2c3"
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서HTTP의 에 대한 경로 작업을 APIs 참조하세요.

- 자세한 API 내용은 명령 참조GetRoute의 섹션을 참조하세요. AWS CLI

### get-routes

다음 코드 예시에서는 get-routes을 사용하는 방법을 보여 줍니다.

AWS CLI

경로 목록을 검색하려면

다음 get-routes 예제에서는 의 모든 API경로 목록을 표시합니다.

```
aws apigatewayv2 get-routes \
    --api-id a1b2c3d4
```

출력:

```
{
    "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"
        }
    ]
```

```
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서HTTP의 에 대한 경로 작업을 APIs 참조하세요.

- 자세한 API 내용은 명령 참조GetRoutes의 섹션을 참조하세요. AWS CLI

### get-stage

다음 코드 예시에서는 get-stage을 사용하는 방법을 보여 줍니다.

AWS CLI

단계에 대한 정보를 검색하려면

다음 get-stage 예제에서는 의 prod 단계에 대한 정보를 표시합니다API.

```
aws apigatewayv2 get-stage \
    --api-id a1b2c3d4 \
    --stage-name prod
```

출력:

```
{
    "CreatedDate": "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": {}
}
```

자세한 내용은 Amazon Gateway 개발자 안내서HTTP의 에 대한 단계 작업을 APIs 참조하세요. API

- 자세한 API 내용은 명령 참조GetStage의 섹션을 참조하세요. AWS CLI

### get-stages

다음 코드 예시에서는 get-stages을 사용하는 방법을 보여 줍니다.

AWS CLI

스테이지 목록을 검색하려면

다음 get-stages 예제에서는 의 모든 단계를 나열API합니다.

```
aws apigatewayv2 get-stages \
    --api-id a1b2c3d4
```

출력:

```
{
    "Items": [
        {
            "ApiGatewayManaged": true,
            "AutoDeploy": true,
            "CreatedDate": "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,
            "CreatedDate": "2020-04-08T00:35:06Z",
            "DefaultRouteSettings": {
                "DetailedMetricsEnabled": false
            },
            "LastUpdatedDate": "2020-04-08T00:35:48Z",
            "RouteSettings": {},
            "StageName": "dev",
            "StageVariables": {
```

```
                "function": "my-dev-function"
            },
            "Tags": {}
        },
        {
            "CreatedDate": "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": {}
        }
    ]
}
```

자세한 내용은 Amazon Gateway 개발자 안내서HTTP의 에 대한 단계 작업을 APIs 참조하세요. API

- 자세한 API 내용은 명령 참조GetStages의 섹션을 참조하세요. AWS CLI

### get-tags

다음 코드 예시에서는 get-tags을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스의 태그 목록을 검색하려면

다음 get-tags 예제에서는 의 모든 태그를 나열API합니다.

```
aws apigatewayv2 get-tags \
    --resource-arn arn:aws:apigateway:us-west-2::/apis/a1b2c3d4
```

출력:

```
{
```

```
    "Tags": {
        "owner": "dev-team",
        "environment": "prod"
    }
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서의 Gateway 리소스 태그 지정을 참조하세요.

API

- 자세한 API 내용은 명령 참조GetTags의 섹션을 참조하세요. AWS CLI

### get-vpc-link

다음 코드 예시에서는 get-vpc-link을 사용하는 방법을 보여 줍니다.

AWS CLI

VPC 링크에 대한 정보를 검색하려면

다음 get-vpc-link 예제에서는 VPC 링크에 대한 정보를 보여줍니다.

```
aws apigatewayv2 get-vpc-link \
    --vpc-link-id abcd123
```

출력:

```
{
    "CreatedDate": "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"
```

```
}
```

자세한 내용은 Amazon Gateway 개발자 안내서HTTP의 에 대한 VPC 링크 작업을 APIs 참조하세요. API

- 자세한 API 내용은 명령 참조GetVpcLink의 섹션을 참조하세요. AWS CLI

### get-vpc-links

다음 코드 예시에서는 get-vpc-links을 사용하는 방법을 보여 줍니다.

AWS CLI

VPC 링크 목록을 검색하려면

다음 get-vpc-links 예제에서는 현재 사용자의 모든 VPC 링크 목록을 표시합니다.

```
aws apigatewayv2 get-vpc-links
```

출력:

```
{
    "Items": [
        {
            "CreatedDate": "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"
        }
        {
            "CreatedDate": "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"
        }
    ]
}
```

자세한 내용은 Amazon Gateway 개발자 안내서HTTP의 에 대한 VPC 링크 작업을 APIs 참조하세요. API

- 자세한 API 내용은 명령 참조GetVpcLinks의 섹션을 참조하세요. AWS CLI

### `import-api`

다음 코드 예시에서는 `import-api`을 사용하는 방법을 보여 줍니다.

AWS CLI

를 가져오려면 HTTP API

다음 `import-api` 예제에서는 라는 OpenAPI 3.0 정의 파일HTTPAPI에서 를 생성합니다`api-definition.yaml`.

```
aws apigatewayv2 import-api \
    --body file://api-definition.yaml
```

`api-definition.yaml`의 콘텐츠:

```
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
```

출력:

```
{
    "ApiEndpoint": "https://a1b2c3d4.execute-api.us-west-2.amazonaws.com",
    "ApiId": "a1b2c3d4",
    "ApiKeySelectionExpression": "$request.header.x-api-key",
    "CreatedDate": "2020-04-08T17:19:38+00:00",
    "Name": "My Lambda API",
    "ProtocolType": "HTTP",
    "RouteSelectionExpression": "$request.method $request.path",
    "Tags": {},
    "Version": "v1.0"
}
```

자세한 내용은 Amazon Gateway 개발자 안내서HTTP의 에 대한 오픈API 정의 작업을 APIs 참조하세요.  API

- 자세한 API 내용은 명령 참조ImportApi의 섹션을 참조하세요. AWS CLI

### `reimport-api`

다음 코드 예시에서는 `reimport-api`을 사용하는 방법을 보여 줍니다.

AWS CLI

를 다시 가져오려면 HTTP API

다음 `reimport-api` 예제에서는 에 지정된 OpenAPI 3.0 정의를 HTTP API 사용하도록 기존 를 업데이트합니다`api-definition.yaml`.

```
aws apigatewayv2 reimport-api \
```

```
    --body file://api-definition.yaml \
    --api-id a1b2c3d4
```

api-definition.yaml의 콘텐츠:

```
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
```

출력:

```
{
    "ApiEndpoint": "https://a1b2c3d4.execute-api.us-west-2.amazonaws.com",
    "ApiId": "a1b2c3d4",
    "ApiKeySelectionExpression": "$request.header.x-api-key",
    "CreatedDate": "2020-04-08T17:19:38+00:00",
    "Name": "My Lambda API",
    "ProtocolType": "HTTP",
    "RouteSelectionExpression": "$request.method $request.path",
    "Tags": {},
    "Version": "v1.0"
}
```

자세한 내용은 Amazon Gateway 개발자 안내서HTTP의 에 대한 개방형API 정의 작업을 APIs 참조하세요.  API

- 자세한 API 내용은 명령 참조ReimportApi의 섹션을 참조하세요. AWS CLI

### tag-resource

다음 코드 예시에서는 tag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 태그를 지정하려면

다음 tag-resource 예제에서는 키 이름과 Department 값이 인 태그를 지정된 Accounting에 추가합니다API.

```
aws apigatewayv2 tag-resource \
    --resource-arn arn:aws:apigateway:us-west-2::/apis/a1b2c3d4 \
    --tags Department=Accounting
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon API Gateway 개발자 안내서의 Gateway 리소스 태그 지정을 참조하세요. API

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### untag-resource

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에서 태그를 제거하려면

다음 untag-resource 예제에서는 키 이름이 Project 및 인 태그를 지정된 Owner에서 제거합니다API.

```
aws apigatewayv2 untag-resource \
    --resource-arn arn:aws:apigateway:us-west-2::/apis/a1b2c3d4 \
    --tag-keys Project Owner
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon API Gateway 개발자 안내서의 Gateway 리소스 태그 지정을 참조하세요. API

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-api-mapping

다음 코드 예시에서는 update-api-mapping을 사용하는 방법을 보여 줍니다.

AWS CLI

API 매핑을 업데이트하려면

다음 update-api-mapping 예제에서는 사용자 지정 도메인 이름에 대한 API 매핑을 변경합니다. 따라서 지정된 API 및 단계에 대한 사용자 지정 도메인 이름을 URL 사용하는 기본이 가 됩니다https://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
```

출력:

```
{
    "ApiId": "a1b2c3d4",
    "ApiMappingId": "0qzs2sy7bh",
    "ApiMappingKey": "dev"
    "Stage": "dev"
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서의 Gateway에서 리전 사용자 지정 도메인 이름 설정을 참조하세요. API

- 자세한 API 내용은 명령 참조UpdateApiMapping의 섹션을 참조하세요. AWS CLI

**update-api**

다음 코드 예시에서는 update-api을 사용하는 방법을 보여 줍니다.

AWS CLI

에 CORS 대해 를 활성화하려면 HTTP API

다음 update-api 예제는 지정된 API의 CORS 구성을 업데이트하여 의 요청을 허용합니다https://www.example.com.

```
aws apigatewayv2 update-api \
```

```
    --api-id a1b2c3d4 \
    --cors-configuration AllowOrigins=https://www.example.com
```

출력:

```
{
    "ApiEndpoint": "https://a1b2c3d4.execute-api.us-west-2.amazonaws.com",
    "ApiId": "a1b2c3d4",
    "ApiKeySelectionExpression": "$request.header.x-api-key",
    "CorsConfiguration": {
        "AllowCredentials": false,
        "AllowHeaders": [
            "header1",
            "header2"
        ],
        "AllowMethods": [
            "GET",
            "OPTIONS"
        ],
        "AllowOrigins": [
            "https://www.example.com"
        ]
    },
    "CreatedDate": "2020-04-08T18:39:37+00:00",
    "Name": "my-http-api",
    "ProtocolType": "HTTP",
    "RouteSelectionExpression": "$request.method $request.path",
    "Tags": {},
    "Version": "v1.0"
}
```

자세한 내용은 Amazon Gateway 개발자 안내서HTTP의 에 CORS 대한 구성을 API 참조하세요. API

- 자세한 API 내용은 명령 참조UpdateApi의 섹션을 참조하세요. AWS CLI

### update-authorizer

다음 코드 예시에서는 update-authorizer을 사용하는 방법을 보여 줍니다.

AWS CLI

권한 부여자를 업데이트하려면

다음 update-authorizer 예제에서는 JWT 권한 부여자의 자격 증명 소스를 라는 헤더로 변경합니다Authorization.

```
aws apigatewayv2 update-authorizer \
    --api-id a1b2c3d4 \
    --authorizer-id a1b2c3 \
    --identity-source '$request.header.Authorization'
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서의 JWT 권한 부여자를 HTTP APIs 사용하여에 대한 액세스 제어를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateAuthorizer의 섹션을 참조하세요. AWS CLI

### update-deployment

다음 코드 예시에서는 update-deployment을 사용하는 방법을 보여 줍니다.

AWS CLI

배포의 설명을 변경하려면

다음 update-deployment 예제에서는 배포의 설명을 업데이트합니다.

```
aws apigatewayv2 update-deployment \
```

```
    --api-id a1b2c3d4 \
    --deployment-id abcdef \
    --description 'Manual deployment to fix integration test failures.'
```

출력:

```
{
    "AutoDeployed": false,
    "CreatedDate": "2020-02-05T16:21:48+00:00",
    "DeploymentId": "abcdef",
    "DeploymentStatus": "DEPLOYED",
    "Description": "Manual deployment to fix integration test failures."
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서 의 GatewayHTTPAPI에서 의 개발을 참조하세요. API

- 자세한 API 내용은 명령 참조UpdateDeployment의 섹션을 참조하세요. AWS CLI

### update-domain-name

다음 코드 예시에서는 update-domain-name을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 도메인 이름을 업데이트하려면

다음 update-domain-name 예제에서는 api.example.com 사용자 지정 도메인 이름에 대한 새 ACM 인증서를 지정합니다.

```
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
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서의 Gateway에서 리전 사용자 지정 도메인 이름 설정을 참조하세요. API

- 자세한 API 내용은 명령 참조UpdateDomainName의 섹션을 참조하세요. AWS CLI

### update-integration

다음 코드 예시에서는 update-integration을 사용하는 방법을 보여 줍니다.

AWS CLI

Lambda 통합을 업데이트하려면

다음 update-integration 예제에서는 지정된 AWS Lambda 함수를 사용하도록 기존 Lambda 통합을 업데이트합니다.

```
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
```

출력:

```
{
    "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
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서의 에 대한 통합 구성 HTTP APIs 및 통합 설정을 WebSocket API 참조하세요.

- 자세한 API 내용은 명령 참조UpdateIntegration의 섹션을 참조하세요. AWS CLI

### update-route

다음 코드 예시에서는 update-route을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 라우팅 통합 업데이트

다음 update-route 예제에서는 지정된 경로의 통합을 업데이트합니다.

```
aws apigatewayv2 update-route \
    --api-id a1b2c3d4 \
    --route-id a1b2c3 \
    --target integrations/a1b2c6
```

출력:

```
{
    "ApiKeyRequired": false,
    "AuthorizationType": "NONE",
    "RouteId": "a1b2c3",
    "RouteKey": "ANY /pets",
    "Target": "integrations/a1b2c6"
}
```

예제 2: 경로에 권한 부여자를 추가하려면

다음 update-route 예제에서는 JWT 권한 부여자를 사용하도록 지정된 경로를 업데이트합니다.

```
aws apigatewayv2 update-route \
    --api-id a1b2c3d4  \
    --route-id a1b2c3  \
    --authorization-type JWT \
    --authorizer-id a1b2c5 \
    --authorization-scopes user.id user.email
```

출력:

```
{
    "ApiKeyRequired": false,
    "AuthorizationScopes": [
        "user.id",
        "user.email"
    ],
    "AuthorizationType": "JWT",
    "AuthorizerId": "a1b2c5",
    "OperationName": "GET HTTP",
    "RequestParameters": {},
    "RouteId": "a1b2c3",
    "RouteKey": "GET /pets",
    "Target": "integrations/a1b2c6"
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서의 JWT 권한 부여자를 HTTP APIs 사용하여에 대한 액세스 제어를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateRoute의 섹션을 참조하세요. AWS CLI

### update-stage

다음 코드 예시에서는 update-stage을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 제한 구성

다음 update-stage 예제에서는 의 지정된 단계 및 경로에 대한 사용자 지정 제한을 구성합니다 API.

```
aws apigatewayv2 update-stage \
```

```
    --api-id a1b2c3d4 \
    --stage-name dev \
    --route-settings '{"GET /pets":
{"ThrottlingBurstLimit":100,"ThrottlingRateLimit":2000}}'
```

출력:

```
{
    "CreatedDate": "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": {}
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서HTTP의 보호 API 섹션을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateStage의 섹션을 참조하세요. AWS CLI

### update-vpc-link

다음 코드 예시에서는 update-vpc-link을 사용하는 방법을 보여 줍니다.

AWS CLI

VPC 링크를 업데이트하려면

다음 update-vpc-link 예제에서는 VPC 링크의 이름을 업데이트합니다. VPC 링크를 생성한 후에는 해당 보안 그룹 또는 서브넷을 변경할 수 없습니다.

```
aws apigatewayv2 update-vpc-link \
    --vpc-link-id abcd123 \
```

```
    --name MyUpdatedVpcLink
```

출력:

```
{
    "CreatedDate": "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"
}
```

자세한 내용은 Amazon Gateway 개발자 안내서HTTP의 에 대한 VPC 링크 작업을 APIs 참조하세요. API

- 자세한 API 내용은 명령 참조UpdateVpcLink의 섹션을 참조하세요. AWS CLI

## API 를 사용한 Gateway Management API 예제 AWS CLI

다음 코드 예제에서는 API Gateway Management와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다API.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### `delete-connection`

다음 코드 예시에서는 `delete-connection`을 사용하는 방법을 보여 줍니다.

AWS CLI

WebSocket 연결을 삭제하려면

다음 `delete-connection` 예제에서는 지정된 에서 클라이언트의 연결을 해제합니다 WebSocket API.

```
aws apigatewaymanagementapi delete-connection \
    --connection-id L0SM9cOFvHcCIhw= \
    --endpoint-url https://aabbccddee.execute-api.us-west-2.amazonaws.com/prod
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon API Gateway 개발자 안내서의 백엔드 서비스에서 @connections 명령 사용을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteConnection의 섹션을 참조하세요. AWS CLI

### `get-connection`

다음 코드 예시에서는 `get-connection`을 사용하는 방법을 보여 줍니다.

AWS CLI

WebSocket 연결에 대한 정보를 가져오려면

다음 `get-connection` 예제에서는 지정된 에 대한 연결을 설명합니다 WebSocket API.

```
aws apigatewaymanagementapi get-connection \
    --connection-id L0SM9cOFvHcCIhw= \
    --endpoint-url https://aabbccddee.execute-api.us-west-2.amazonaws.com/prod
```

출력:

```
{
    "ConnectedAt": "2020-04-30T20:10:33.236Z",
    "Identity": {
```

```
        "SourceIp": "192.0.2.1"
    },
    "LastActiveAt": "2020-04-30T20:10:42.997Z"
}
```

자세한 내용은 Amazon API Gateway 개발자 안내서의 백엔드 서비스에서 @connections 명령 사용을 참조하세요.

- 자세한 API 내용은 명령 참조GetConnection의 섹션을 참조하세요. AWS CLI

**post-to-connection**

다음 코드 예시에서는 post-to-connection을 사용하는 방법을 보여 줍니다.

AWS CLI

WebSocket 연결로 데이터를 보내려면

다음 post-to-connection 예제에서는 지정된 에 연결된 클라이언트에 메시지를 보냅니다 WebSocket API.

```
aws apigatewaymanagementapi post-to-connection \
    --connection-id L0SM9cOFvHcCIhw= \
    --data "Hello from API Gateway!" \
    --endpoint-url https://aabbccddee.execute-api.us-west-2.amazonaws.com/prod
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon API Gateway 개발자 안내서의 백엔드 서비스에서 @connections 명령 사용을 참조하세요.

- 자세한 API 내용은 명령 참조PostToConnection의 섹션을 참조하세요. AWS CLI

## 를 사용한 App Mesh 예제 AWS CLI

다음 코드 예제에서는 App Mesh AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `create-mesh`

다음 코드 예시에서는 `create-mesh`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 새 서비스 메시 생성

다음 `create-mesh` 예제에서는 서비스 메시를 생성합니다.

```
aws appmesh create-mesh \
    --mesh-name app1
```

출력:

```
{
    "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"
        }
    }
}
```

예제 2: 여러 태그가 있는 새 서비스 메시 생성

다음 create-mesh 예제에서는 여러 태그가 있는 서비스 메시를 생성합니다.

```
aws appmesh create-mesh \
    --mesh-name app2 \
    --tags key=key1,value=value1 key=key2,value=value2 key=key3,value=value3
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 AWS App Mesh 사용 설명서의 Service Meshes를 참조하세요.

- 자세한 API 내용은 명령 참조CreateMesh의 섹션을 참조하세요. AWS CLI

### create-route

다음 코드 예시에서는 create-route을 사용하는 방법을 보여 줍니다.

AWS CLI

새 gRPC 라우팅을 생성하려면

다음 create-route 예제에서는 JSON 입력 파일을 사용하여 gRPC 라우팅을 생성합니다. GRPC 123으로 시작하는 메타데이터가 있는 트래픽은 라는 가상 노드로 라우팅됩니다serviceBgrpc. 라우팅의 대상과 통신을 시도할 때 특정 gRPCHTTP, 또는 TCP 실패가 있는 경우 라우팅이 세 번 재시도됩니다. 각 재시도 사이에는 15초의 지연이 있습니다.

```
aws appmesh create-route \
    --cli-input-json file://create-route-grpc.json
```

create-route-grpc.json의 콘텐츠:

```
{
    "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"
}
```

출력:

```
{
    "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"
    }
}
```

새 HTTP 또는 HTTP/2 경로를 생성하려면

다음 `create-route` 예제에서는 JSON 입력 파일을 사용하여 HTTP/2 경로를 생성합니다. HTTP 경로를 생성하려면 http2Route를 사양 httpRoute 에 따라 로 바꿉니다. 헤더 값이 123으로 시작하는 URL 접두사로 지정된 모든 HTTP/2 트래픽은 serviceBhttp2라는 가상 노드로 라우팅됩니다. 경로의 대상과 통신을 시도할 때 특정 HTTP 또는 TCP 장애가 있는 경우 경로가 3회 재시도됩니다. 각 재시도 사이에는 15초의 지연이 있습니다.

```
aws appmesh create-route \
    --cli-input-json file://create-route-http2.json
```

`create-route-http2.json`의 콘텐츠:

```
{
    "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"
}
```

출력:

```
{
    "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"
    }
}
```

새 TCP 경로를 생성하려면

다음 `create-route` 예제에서는 JSON 입력 파일을 사용하여 TCP 라우팅을 생성합니다. 트래픽의 75%가 라는 가상 노드로 라우팅serviceBtcp되고 트래픽의 25%가 serviceBv2tcp라는 가상 노드로 라우팅됩니다. 다양한 대상에 대해 다양한 가중치를 지정하는 것은 새 버전의 애플리케이션을 배포하는 효과적인 방법입니다. 최종적으로 모든 트래픽의 100%가 애플리케이션의 새 버전이 있는 대상으로 라우팅되도록 가중치를 조정할 수 있습니다.

```
aws appmesh create-route \
    --cli-input-json file://create-route-tcp.json
```

create-route-tcp.json의 내용:

```
{
    "meshName": "apps",
    "routeName": "tcpRoute",
    "spec": {
        "priority": 300,
        "tcpRoute": {
            "action": {
                "weightedTargets": [
                    {
                        "virtualNode": "serviceBtcp",
                        "weight": 75
```

```
                    },
                    {
                        "virtualNode": "serviceBv2tcp",
                        "weight": 25
                    }
                ]
            }
        }
    },
    "virtualRouterName": "serviceBtcp"
}
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS App Mesh 사용 설명서의 라우팅을 참조하세요.

- 자세한 API 내용은 명령 참조CreateRoute의 섹션을 참조하세요. AWS CLI

### create-virtual-gateway

다음 코드 예시에서는 create-virtual-gateway을 사용하는 방법을 보여 줍니다.

AWS CLI

새 가상 게이트웨이를 생성하려면

다음 create-virtual-gateway 예제에서는 JSON 입력 파일을 사용하여 포트 9080을 HTTP 사용하기 위한 리스너를 사용하여 가상 게이트웨이를 생성합니다.

```
aws appmesh create-virtual-gateway \
    --mesh-name meshName \
    --virtual-gateway-name virtualGatewayName \
    --cli-input-json file://create-virtual-gateway.json
```

create-virtual-gateway.json의 콘텐츠:

```
{
    "spec": {
      "listeners": [
        {
          "portMapping": {
            "port": 9080,
            "protocol": "http"
          }
        }
      ]
    }
}
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS App Mesh 사용 설명서의 Virtual Gateways를 참조하세요.

- 자세한 API 내용은 명령 참조CreateVirtualGateway의 섹션을 참조하세요. AWS CLI

### `create-virtual-node`

다음 코드 예시에서는 `create-virtual-node`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 가 검색DNS에 사용하는 새 가상 노드를 생성하려면

다음 create-virtual-node 예제에서는 JSON 입력 파일을 사용하여 가 서비스 검색에 사용하는 가상 노드DNS를 생성합니다.

```
aws appmesh create-virtual-node \
    --cli-input-json file://create-virtual-node-dns.json
```

create-virtual-node-dns.json의 콘텐츠:

```
{
    "meshName": "app1",
    "spec": {
        "listeners": [
            {
                "portMapping": {
                    "port": 80,
                    "protocol": "http"
                }
            }
        ],
        "serviceDiscovery": {
            "dns": {
                "hostname": "serviceBv1.svc.cluster.local"
            }
        }
    },
    "virtualNodeName": "vnServiceBv1"
}
```

출력:

```
{
    "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"
    }
}
```

예제 2: 검색에 AWS Cloud Map을 사용하는 새 가상 노드 생성

다음 `create-virtual-node` 예제에서는 JSON 입력 파일을 사용하여 서비스 검색에 AWS Cloud Map을 사용하는 가상 노드를 생성합니다.

```
aws appmesh create-virtual-node \
    --cli-input-json file://create-virtual-node-cloud-map.json
```

`create-virtual-node-cloud-map.json`의 콘텐츠:

```
{
    "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"
}
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS App Mesh 사용 설명서의 가상 노드를 참조하세요.

- 자세한 API 내용은 명령 참조CreateVirtualNode의 섹션을 참조하세요. AWS CLI

### `create-virtual-router`

다음 코드 예시에서는 `create-virtual-router`을 사용하는 방법을 보여 줍니다.

AWS CLI

새 가상 라우터를 생성하려면

다음 `create-virtual-router` 예제에서는 JSON 입력 파일을 사용하여 포트 80을 HTTP 사용하기 위한 리스너가 있는 가상 라우터를 생성합니다.

```
aws appmesh create-virtual-router \
    --cli-input-json file://create-virtual-router.json
```

create-virtual-router.json의 콘텐츠:

```
{
    "meshName": "app1",
    "spec": {
        "listeners": [
            {
                "portMapping": {
                    "port": 80,
                    "protocol": "http"
                }
            }
        ]
    },
    "virtualRouterName": "vrServiceB"
}
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS App Mesh 사용 설명서의 가상 라우터를 참조하세요.

- 자세한 API 내용은 명령 참조CreateVirtualRouter의 섹션을 참조하세요. AWS CLI

### `create-virtual-service`

다음 코드 예시에서는 `create-virtual-service`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 가상 노드 공급자를 사용하여 새 가상 서비스를 생성하려면

다음 `create-virtual-service` 예제에서는 JSON 입력 파일을 사용하여 가상 노드 공급자를 사용하여 가상 서비스를 생성합니다.

```
aws appmesh create-virtual-service \
    --cli-input-json file://create-virtual-service-virtual-node.json
```

`create-virtual-service-virtual-node.json`의 콘텐츠:

```
{
    "meshName": "app1",
    "spec": {
        "provider": {
            "virtualNode": {
                "virtualNodeName": "vnServiceA"
            }
        }
    },
    "virtualServiceName": "serviceA.svc.cluster.local"
}
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS App Mesh 사용 설명서의 가상 노드를 참조하세요.

예제 2: 가상 라우터 공급자를 사용하여 새 가상 서비스를 생성하려면

다음 `create-virtual-service` 예제에서는 JSON 입력 파일을 사용하여 가상 라우터 공급자를 사용하여 가상 서비스를 생성합니다.

```
aws appmesh create-virtual-service \
    --cli-input-json file://create-virtual-service-virtual-router.json
```

`create-virtual-service-virtual-router.json`의 콘텐츠:

```
{
    "meshName": "app1",
    "spec": {
        "provider": {
            "virtualRouter": {
```

```
                "virtualRouterName": "vrServiceB"
            }
        }
    },
    "virtualServiceName": "serviceB.svc.cluster.local"
}
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS App Mesh 사용 설명서의 Virtual Services<https://docs.aws.amazon.com/app-mesh/latest/userguide/virtual_services.html>을 참조하세요.

- 자세한 API 내용은 명령 참조CreateVirtualService의 섹션을 참조하세요. AWS CLI

### `delete-mesh`

다음 코드 예시에서는 `delete-mesh`을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스 메시를 삭제하려면

다음 delete-mesh 예제에서는 지정된 서비스 메시를 삭제합니다.

```
aws appmesh delete-mesh \
    --mesh-name app1
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 AWS App Mesh 사용 설명서의 Service Meshes를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteMesh의 섹션을 참조하세요. AWS CLI

## delete-route

다음 코드 예시에서는 delete-route을 사용하는 방법을 보여 줍니다.

AWS CLI

라우팅을 삭제하려면

다음 delete-route 예제에서는 지정된 경로를 삭제합니다.

```
aws appmesh delete-route \
    --mesh-name app1 \
    --virtual-router-name vrServiceB \
    --route-name toVnServiceB-weighted
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS App Mesh 사용 설명서의 라우팅을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteRoute의 섹션을 참조하세요. AWS CLI

### delete-virtual-node

다음 코드 예시에서는 delete-virtual-node을 사용하는 방법을 보여 줍니다.

AWS CLI

가상 노드를 삭제하려면

다음 delete-virtual-node 예제에서는 지정된 가상 노드를 삭제합니다.

```
aws appmesh delete-virtual-node \
    --mesh-name app1 \
    --virtual-node-name vnServiceBv2
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS App Mesh 사용 설명서의 가상 노드를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteVirtualNode의 섹션을 참조하세요. AWS CLI

### delete-virtual-router

다음 코드 예시에서는 delete-virtual-router을 사용하는 방법을 보여 줍니다.

AWS CLI

가상 라우터를 삭제하려면

다음 delete-virtual-router 예제에서는 지정된 가상 라우터를 삭제합니다.

```
aws appmesh delete-virtual-router \
    --mesh-name app1 \
    --virtual-router-name vrServiceB
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS App Mesh 사용 설명서의 가상 라우터를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteVirtualRouter의 섹션을 참조하세요. AWS CLI

### delete-virtual-service

다음 코드 예시에서는 delete-virtual-service을 사용하는 방법을 보여 줍니다.

AWS CLI

가상 서비스를 삭제하려면

다음 delete-virtual-service 예제에서는 지정된 가상 서비스를 삭제합니다.

```
aws appmesh delete-virtual-service \
    --mesh-name app1 \
    --virtual-service-name serviceB.svc.cluster.local
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS App Mesh 사용 설명서의 Virtual Service를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteVirtualService의 섹션을 참조하세요. AWS CLI

### describe-mesh

다음 코드 예시에서는 describe-mesh을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스 메시를 설명하려면

다음 describe-mesh 예제에서는 지정된 서비스 메시에 대한 세부 정보를 반환합니다.

```
aws appmesh describe-mesh \
    --mesh-name app1
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 AWS App Mesh 사용 설명서의 Service Meshes를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeMesh의 섹션을 참조하세요. AWS CLI

### describe-route

다음 코드 예시에서는 describe-route을 사용하는 방법을 보여 줍니다.

AWS CLI

경로를 설명하려면

다음 describe-route 예제에서는 지정된 경로에 대한 세부 정보를 반환합니다.

```
aws appmesh describe-route \
    --mesh-name app1 \
    --virtual-router-name vrServiceB \
    --route-name toVnServiceB-weighted
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS App Mesh 사용 설명서의 라우팅을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeRoute의 섹션을 참조하세요. AWS CLI

### describe-virtual-node

다음 코드 예시에서는 describe-virtual-node을 사용하는 방법을 보여 줍니다.

AWS CLI

가상 노드를 설명하려면

다음 describe-virtual-node 예제에서는 지정된 가상 노드에 대한 세부 정보를 반환합니다.

```
aws appmesh describe-virtual-node \
    --mesh-name app1 \
    --virtual-node-name vnServiceBv1
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS App Mesh 사용 설명서의 가상 노드를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeVirtualNode의 섹션을 참조하세요. AWS CLI

### describe-virtual-router

다음 코드 예시에서는 describe-virtual-router을 사용하는 방법을 보여 줍니다.

AWS CLI

가상 라우터를 설명하려면

다음 `describe-virtual-router` 예제에서는 지정된 가상 라우터에 대한 세부 정보를 반환합니다.

```
aws appmesh describe-virtual-router \
    --mesh-name app1 \
    --virtual-router-name vrServiceB
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS App Mesh 사용 설명서의 가상 라우터를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeVirtualRouter의 섹션을 참조하세요. AWS CLI

### describe-virtual-service

다음 코드 예시에서는 describe-virtual-service을 사용하는 방법을 보여 줍니다.

AWS CLI

가상 서비스를 설명하려면

다음 describe-virtual-service 예제에서는 지정된 가상 서비스에 대한 세부 정보를 반환합니다.

```
aws appmesh describe-virtual-service \
    --mesh-name app1 \
    --virtual-service-name serviceB.svc.cluster.local
```

출력:

```
{
    "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"
    }
```

```
}
```

자세한 내용은 AWS App Mesh 사용 설명서의 가상 서비스를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeVirtualService의 섹션을 참조하세요. AWS CLI

### `list-meshes`

다음 코드 예시에서는 `list-meshes`을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스 메시를 나열하려면

다음 `list-meshes` 예제에서는 현재 AWS 리전의 모든 서비스 메시를 나열합니다.

```
aws appmesh list-meshes
```

출력:

```
{
    "meshes": [
        {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1",
            "meshName": "app1"
        }
    ]
}
```

자세한 내용은 AWS App Mesh 사용 설명서의 Service Meshes를 참조하세요.

- 자세한 API 내용은 명령 참조ListMeshes의 섹션을 참조하세요. AWS CLI

### `list-routes`

다음 코드 예시에서는 `list-routes`을 사용하는 방법을 보여 줍니다.

AWS CLI

경로를 나열하려면

다음 `list-routes` 예제에서는 지정된 가상 라우터의 모든 경로를 나열합니다.

```
aws appmesh list-routes \
    --mesh-name app1 \
    --virtual-router-name vrServiceB
```

출력:

```
{
    "routes": [
        {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/
vrServiceB/route/toVnServiceB",
            "meshName": "app1",
            "routeName": "toVnServiceB-weighted",
            "virtualRouterName": "vrServiceB"
        }
    ]
}
```

자세한 내용은 AWS App Mesh 사용 설명서의 라우팅을 참조하세요.

- 자세한 API 내용은 명령 참조ListRoutes의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스의 태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 지정된 리소스에 할당된 모든 태그를 나열합니다.

```
aws appmesh list-tags-for-resource \
    --resource-arn arn:aws:appmesh:us-east-1:123456789012:mesh/app1
```

출력:

```
{
    "tags": [
        {
            "key": "key1",
            "value": "value1"
```

```
        },
        {
            "key": "key2",
            "value": "value2"
        },
        {
            "key": "key3",
            "value": "value3"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### `list-virtual-nodes`

다음 코드 예시에서는 `list-virtual-nodes`을 사용하는 방법을 보여 줍니다.

AWS CLI

가상 노드를 나열하려면

다음 `list-virtual-nodes` 예제에서는 지정된 서비스 메시의 모든 가상 노드를 나열합니다.

```
aws appmesh list-virtual-nodes \
    --mesh-name app1
```

출력:

```
{
    "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"
```

```
        }
    ]
}
```

자세한 내용은 AWS App Mesh 사용 설명서의 가상 노드를 참조하세요.

- 자세한 API 내용은 명령 참조ListVirtualNodes의 섹션을 참조하세요. AWS CLI

### `list-virtual-routers`

다음 코드 예시에서는 `list-virtual-routers`을 사용하는 방법을 보여 줍니다.

AWS CLI

가상 라우터를 나열하려면

다음 `list-virtual-routers` 예제에서는 지정된 서비스 메시의 모든 가상 라우터를 나열합니다.

```
aws appmesh list-virtual-routers \
    --mesh-name app1
```

출력:

```
{
    "virtualRouters": [
        {
            "arn": "arn:aws:appmesh:us-east-1:123456789012:mesh/app1/virtualRouter/
vrServiceB",
            "meshName": "app1",
            "virtualRouterName": "vrServiceB"
        }
    ]
}
```

자세한 내용은 AWS App Mesh 사용 설명서의 가상 라우터를 참조하세요.

- 자세한 API 내용은 명령 참조ListVirtualRouters의 섹션을 참조하세요. AWS CLI

### `list-virtual-services`

다음 코드 예시에서는 `list-virtual-services`을 사용하는 방법을 보여 줍니다.

AWS CLI

가상 서비스를 나열하려면

다음 `list-virtual-services` 예제에서는 지정된 서비스 메시의 모든 가상 서비스를 나열합니다.

```
aws appmesh list-virtual-services \
    --mesh-name app1
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS App Mesh 사용 설명서의 가상 서비스를 참조하세요.

- 자세한 API 내용은 명령 참조ListVirtualServices의 섹션을 참조하세요. AWS CLI

### `tag-resource`

다음 코드 예시에서는 `tag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 태그를 지정하려면

다음 `tag-resource` 예제에서는 `value1` 지정된 리소스에 값이 `key1` 인 태그를 추가합니다.

```
aws appmesh tag-resource \
    --resource-arn arn:aws:appmesh:us-east-1:123456789012:mesh/app1 \
    --tags key=key1,value=value1
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### untag-resource

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스의 태그를 해제하려면

다음 untag-resource 예제에서는 지정된 리소스key1에서 키가 있는 태그를 제거합니다.

```
aws appmesh untag-resource \
    --resource-arn arn:aws:appmesh:us-east-1:123456789012:mesh/app1 \
    --tag-keys key1
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-mesh

다음 코드 예시에서는 update-mesh을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스 메시를 업데이트하려면

다음 update-mesh 예제에서는 JSON 입력 파일을 사용하여 서비스 메시를 업데이트하여 모든 외부 송신 트래픽이 Envoy 프록시를 통해 터치되지 않고 전달되도록 허용합니다.

```
aws appmesh update-mesh \
    --cli-input-json file://update-mesh.json
```

update-mesh.json의 콘텐츠:

```
{
    "meshName": "app1",
    "spec": {
        "egressFilter": {
            "type": "ALLOW_ALL"
        }
    }
}
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 AWS App Mesh 사용 설명서의 Service Meshes를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateMesh의 섹션을 참조하세요. AWS CLI

### update-route

다음 코드 예시에서는 update-route을 사용하는 방법을 보여 줍니다.

AWS CLI

경로를 업데이트하려면

다음 update-route 예제에서는 JSON 입력 파일을 사용하여 경로의 가중치를 업데이트합니다.

```
aws appmesh update-route \
    --cli-input-json file://update-route-weighted.json
```

update-route-weighted.json의 콘텐츠:

```
{
    "meshName": "app1",
    "routeName": "toVnServiceB-weighted",
    "spec": {
        "httpRoute": {
            "action": {
                "weightedTargets": [
                    {
                        "virtualNode": "vnServiceBv1",
                        "weight": 80
                    },
                    {
                        "virtualNode": "vnServiceBv2",
                        "weight": 20
                    }
                ]
            },
            "match": {
                "prefix": "/"
            }
        }
    },
    "virtualRouterName": "vrServiceB"
}
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS App Mesh 사용 설명서의 라우팅을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateRoute의 섹션을 참조하세요. AWS CLI

### update-virtual-node

다음 코드 예시에서는 update-virtual-node을 사용하는 방법을 보여 줍니다.

AWS CLI

가상 노드를 업데이트하려면

다음 update-virtual-node 예제에서는 JSON 입력 파일을 사용하여 가상 노드에 상태 확인을 추가합니다.

```
aws appmesh update-virtual-node \
    --cli-input-json file://update-virtual-node.json
```

update-virtual-node.json의 콘텐츠:

```
{
    "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"
}
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS App Mesh 사용 설명서의 가상 노드를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateVirtualNode의 섹션을 참조하세요. AWS CLI

### update-virtual-router

다음 코드 예시에서는 update-virtual-router을 사용하는 방법을 보여 줍니다.

AWS CLI

가상 라우터를 업데이트하려면

다음 update-virtual-router 예제에서는 JSON 입력 파일을 사용하여 가상 라우터 리스너 포트를 업데이트합니다.

```
aws appmesh update-virtual-router \
    --cli-input-json file://update-virtual-router.json
```

update-virtual-router.json의 콘텐츠:

```
{
    "meshName": "app1",
    "spec": {
        "listeners": [
            {
                "portMapping": {
                    "port": 8080,
                    "protocol": "http"
                }
            }
        ]
    },
    "virtualRouterName": "vrServiceB"
}
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS App Mesh 사용 설명서의 가상 라우터를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateVirtualRouter의 섹션을 참조하세요. AWS CLI

### update-virtual-service

다음 코드 예시에서는 update-virtual-service을 사용하는 방법을 보여 줍니다.

AWS CLI

가상 서비스를 업데이트하려면

다음 update-virtual-service 예제에서는 JSON 입력 파일을 사용하여 가상 라우터 공급자를 사용하도록 가상 서비스를 업데이트합니다.

```
aws appmesh update-virtual-service \
    --cli-input-json file://update-virtual-service.json
```

update-virtual-service.json의 콘텐츠:

```
{
    "meshName": "app1",
    "spec": {
        "provider": {
            "virtualRouter": {
```

```
                "virtualRouterName": "vrServiceA"
            }
        }
    },
    "virtualServiceName": "serviceA.svc.cluster.local"
}
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS App Mesh 사용 설명서의 가상 서비스를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateVirtualService의 섹션을 참조하세요. AWS CLI

## 를 사용한 App Runner 예제 AWS CLI

다음 코드 예제에서는 App Runner AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `associate-custom-domain`

다음 코드 예시에서는 `associate-custom-domain`을 사용하는 방법을 보여 줍니다.

AWS CLI

도메인 이름과 www 하위 도메인을 서비스에 연결하려면

다음 `associate-custom-domain` 예제에서는 사용자가 제어하는 사용자 지정 도메인 이름을 App Runner 서비스와 연결합니다. 도메인 이름은 특수 사례 하위 도메인 를 `example.com`포함한 루트 도메인 입니다`www.example.com`.

```
aws apprunner associate-custom-domain \
    --cli-input-json file://input.json
```

`input.json`의 콘텐츠:

```
{
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
    "DomainName": "example.com",
    "EnableWWWSubdomain": true
}
```

출력:

```
{
    "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"
}
```

- 자세한 API 내용은 명령 참조AssociateCustomDomain의 섹션을 참조하세요. AWS CLI

### `create-auto-scaling-configuration`

다음 코드 예시에서는 `create-auto-scaling-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

고가용성 Auto Scaling 구성을 생성하려면

다음 `create-auto-scaling-configuration` 예제에서는 를 `5MinSize`로 설정하여 고가용성에 최적화된 자동 조정 구성을 생성합니다. 이 구성을 사용하면 App Runner는 AWS 리전에 따라 최대 5개까지 가능한 대부분의 가용 영역에 서비스 인스턴스를 분산하려고 시도합니다.

호출은 다른 설정이 기본값으로 설정된 `AutoScalingConfiguration` 객체를 반환합니다. 이 예제에서는 라는 구성을 생성하기 위한 첫 번째 호출입니다`high-availability`. 개정은 1로 설정되며 최신 개정입니다.

```
aws apprunner create-auto-scaling-configuration \
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
    "AutoScalingConfigurationName": "high-availability",
    "MinSize": 5
}
```

출력:

```
{
    "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
    }
}
```

- 자세한 API 내용은 명령 참조CreateAutoScalingConfiguration의 섹션을 참조하세요. AWS CLI

### create-connection

다음 코드 예시에서는 create-connection을 사용하는 방법을 보여 줍니다.

AWS CLI

GitHub 연결을 생성하려면

다음 create-connection 예제에서는 프라이빗 GitHub 코드 리포지토리에 대한 연결을 생성합니다. 성공적인 호출 후 연결 상태는 입니다PENDING_HANDSHAKE. 공급자의 인증 핸드셰이크가 아직 발생하지 않았기 때문입니다. App Runner 콘솔을 사용하여 핸드셰이크를 완료합니다.

```
aws apprunner create-connection \
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
    "ConnectionName": "my-github-connection",
    "ProviderType": "GITHUB"
}
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 App Runner 개발자 안내서의 App Runner 연결 관리를 참조하세요. AWS

- 자세한 API 내용은 명령 참조CreateConnection의 섹션을 참조하세요. AWS CLI

### create-service

다음 코드 예시에서는 create-service을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 소스 코드 리포지토리 서비스 생성

다음 create-service 예제에서는 Python 소스 코드 리포지토리를 기반으로 App Runner 서비스를 생성합니다.

```
aws apprunner create-service \
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
    "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"
    }
}
```

출력:

```
{
    "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"
        }
    }
}
```

예제 2: 소스 코드 리포지토리 서비스 생성

다음 `create-service` 예제에서는 Python 소스 코드 리포지토리를 기반으로 App Runner 서비스를 생성합니다.

```
aws apprunner create-service \
    --cli-input-json file://input.json
```

`input.json`의 콘텐츠:

```
{
    "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"
```

```
    }
}
```

출력:

```
{
    "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"
        }
    }
}
```

예제 3: 소스 이미지 리포지토리 서비스 생성

다음 `create-service` 예제에서는 Elastic Container Registry()에 저장된 이미지를 기반으로 App Runner 서비스를 생성합니다ECR.

```
aws apprunner create-service \
    --cli-input-json file://input.json
```

`input.json`의 콘텐츠:

```
{
    "ServiceName": "golang-container-app",
    "SourceConfiguration": {
        "AuthenticationConfiguration": {
            "AccessRoleArn": "arn:aws:iam::123456789012:role/my-ecr-role"
        },
        "AutoDeploymentsEnabled": true,
        "ImageRepository": {
            "ImageIdentifier": "123456789012.dkr.ecr.us-east-1.amazonaws.com/golang-
app:latest",
            "ImageConfiguration": {
                "Port": "8080",
                "RuntimeEnvironmentVariables": [
                    {
                        "NAME": "Jane"
                    }
                ]
            },
            "ImageRepositoryType": "ECR"
        }
    },
    "InstanceConfiguration": {
        "CPU": "1 vCPU",
        "Memory": "3 GB"
```

```
    }
}
```

출력:

```
{
    "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"
        }
    }
}
```

- 자세한 API 내용은 명령 참조CreateService의 섹션을 참조하세요. AWS CLI

### delete-auto-scaling-configuration

다음 코드 예시에서는 delete-auto-scaling-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: Auto Scaling 구성의 최신 활성 개정을 삭제하려면

다음 delete-auto-scaling-configuration 예제에서는 App Runner Auto Scaling 구성의 최신 활성 개정을 삭제합니다. 최신 활성 개정을 삭제하려면 개정 구성 요소 없이 구성 이름으로 끝나는 Amazon 리소스 이름(ARN)을 지정합니다.

이 예제에서는 이 작업 앞에 두 가지 개정이 있습니다. 따라서 개정 2(최신)가 삭제됩니다. 하지만 이제 가 표시됩니다. 삭제된 후에는 더 이상 최신 활성 개정이 아니기 "Latest": false때문입니다.

```
aws apprunner delete-auto-scaling-configuration \
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
    "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/high-availability"
}
```

출력:

```
{
    "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
    }
}
```

예제 2: Auto Scaling 구성의 특정 개정을 삭제하려면

다음 `delete-auto-scaling-configuration` 예제에서는 App Runner Auto Scaling 구성의 특정 개정을 삭제합니다. 특정 개정을 삭제하려면 개정 번호가 ARN 포함된 를 지정합니다.

이 예제에서는 이 작업 앞에 몇 가지 개정이 있습니다. 작업은 개정 를 삭제합니다1.

```
aws apprunner delete-auto-scaling-configuration \
    --cli-input-json file://input.json
```

`input.json`의 콘텐츠:

```
{
    "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/high-availability/1"
}
```

출력:

```
{
    "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
    }
}
```

- 자세한 API 내용은 명령 참조DeleteAutoScalingConfiguration의 섹션을 참조하세요. AWS CLI

### delete-connection

다음 코드 예시에서는 delete-connection을 사용하는 방법을 보여 줍니다.

AWS CLI

연결을 삭제하려면

다음 delete-connection 예제에서는 App Runner 연결을 삭제합니다. 성공적인 호출 후 연결 상태는 입니다DELETED. 이는 연결을 더 이상 사용할 수 없기 때문입니다.

```
aws apprunner delete-connection \
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
    "ConnectionArn": "arn:aws:apprunner:us-east-1:123456789012:connection/my-github-connection"
}
```

출력:

```
{
    "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"
    }
}
```

- 자세한 API 내용은 명령 참조DeleteConnection의 섹션을 참조하세요. AWS CLI

### delete-service

다음 코드 예시에서는 delete-service을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스를 삭제하는 방법

다음 `delete-service` 예제에서는 App Runner 서비스를 삭제합니다.

```
aws apprunner delete-service \
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

출력:

```
{
    "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"
        }
    }
}
```

- 자세한 API 내용은 명령 참조DeleteService의 섹션을 참조하세요. AWS CLI

### `describe-auto-scaling-configuration`

다음 코드 예시에서는 `describe-auto-scaling-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: Auto Scaling 구성의 최신 활성 개정 설명

다음 `describe-auto-scaling-configuration` 예제에서는 App Runner Auto Scaling 구성의 최신 활성 개정에 대한 설명을 제공합니다. 최신 활성 개정을 설명하려면 개정 구성 요소 없이 구성 이름으로 끝ARN나는 를 지정합니다.

이 예제에서는 두 가지 개정이 있습니다. 따라서 개정2(최신)이 설명되어 있습니다. 결과 객체에 이 표시됩니다`"Latest": true`.

```
aws apprunner describe-auto-scaling-configuration \
```

```
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
    "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/high-availability"
}
```

출력:

```
{
    "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
    }
}
```

예제 2: Auto Scaling 구성의 특정 개정을 설명하는 방법

다음 describe-auto-scaling-configuration 예제에서는 App Runner Auto Scaling 구성의 특정 개정에 대한 설명을 제공합니다. 특정 개정을 설명하려면 개정 번호가 ARN 포함된 를 지정합니다.

이 예제에서는 몇 가지 개정이 존재하고 개정1이 쿼리됩니다. 결과 객체에 이 표시됩니다"Latest": false.

```
aws apprunner describe-auto-scaling-configuration \
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
    "AutoScalingConfigurationArn": "arn:aws:apprunner:us-
east-1:123456789012:autoscalingconfiguration/high-availability/1"
}
```

출력:

```
{
    "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
    }
}
```

- 자세한 API 내용은 명령 참조DescribeAutoScalingConfiguration의 섹션을 참조하세요. AWS CLI

### describe-custom-domains

다음 코드 예시에서는 describe-custom-domains을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스와 연결된 사용자 지정 도메인 이름에 대한 설명을 가져오려면

다음 describe-custom-domains 예제에서는 App Runner 서비스와 연결된 사용자 지정 도메인 이름에 대한 설명과 상태를 가져옵니다.

```
aws apprunner describe-custom-domains \
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
    "DomainName": "example.com",
    "EnableWWWSubdomain": true
}
```

출력:

```
{
    "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"
}
```

- 자세한 API 내용은 명령 참조DescribeCustomDomains의 섹션을 참조하세요. AWS CLI

### `describe-service`

다음 코드 예시에서는 `describe-service`을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스를 설명하려면

다음 `describe-service` 예제에서는 App Runner 서비스에 대한 설명을 제공합니다.

```
aws apprunner describe-service \
    --cli-input-json file://input.json
```

`input.json`의 콘텐츠:

```
{
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

출력:

```
{
    "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"
        }
    }
}
```

- 자세한 API 내용은 명령 참조DescribeService의 섹션을 참조하세요. AWS CLI

### `disassociate-custom-domain`

다음 코드 예시에서는 `disassociate-custom-domain`을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스에서 도메인 이름을 연결 해제하려면

다음 disassociate-custom-domain 예제에서는 App Runner 서비스example.com에서 도메인 연결을 해제합니다. 또한 호출은 루트 도메인과 www.example.com 연결된 하위 도메인의 연결을 해제합니다.

```
aws apprunner disassociate-custom-domain \
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
    "DomainName": "example.com"
}
```

출력:

```
{
    "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"
}
```

- 자세한 API 내용은 명령 참조DisassociateCustomDomain의 섹션을 참조하세요. AWS CLI

### `list-auto-scaling-configurations`

다음 코드 예시에서는 `list-auto-scaling-configurations`을 사용하는 방법을 보여 줍니다.

AWS CLI

App Runner Auto Scaling 구성의 페이지 매김 목록을 가져오려면

다음 `list-auto-scaling-configurations` 예제에서는 AWS 계정의 모든 App Runner Auto Scaling 구성을 나열합니다. 각 응답에 최대 5개의 자동 조정 구성이 나열됩니다. `AutoScalingConfigurationName` 및 `LatestOnly` 는 지정되지 않습니다. 기본값으로 인해 모든 활성 구성의 최신 개정이 나열됩니다.

이 예제에서는 응답에 두 개의 결과가 포함되어 있고 추가 결과가 없으므로 반환`NextToken`되지 않습니다.

```
aws apprunner list-auto-scaling-configurations \
    --cli-input-json file://input.json
```

`input.json`의 콘텐츠:

```
{
    "MaxResults": 5
}
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListAutoScalingConfigurations의 섹션을 참조하세요. AWS CLI

### `list-connections`

다음 코드 예시에서는 `list-connections`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 모든 연결을 나열하려면

다음 `list-connections` 예제에서는 AWS 계정의 모든 App Runner 연결을 나열합니다.

```
aws apprunner list-connections
```

출력:

```
{
    "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"
        }
    ]
}
```

예제 2: 이름별로 연결을 나열하려면

다음 `list-connections` 예제에서는 이름별로 연결을 나열합니다.

```
aws apprunner list-connections \
    --cli-input-json file://input.json
```

`input.json`의 콘텐츠:

```
{
    "ConnectionName": "my-github-org-connection"
}
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListConnections의 섹션을 참조하세요. AWS CLI

### `list-operations`

다음 코드 예시에서는 `list-operations`을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스에서 발생한 작업을 나열하려면

다음 `list-operations` 예제에서는 지금까지 App Runner 서비스에서 발생한 모든 작업을 나열합니다. 이 예제에서는 서비스가 새 서비스이며 유형의 단일 작업만 `CREATE_SERVICE` 발생했습니다.

```
aws apprunner list-operations \
    --cli-input-json file://input.json
```

`input.json`의 콘텐츠:

```
{
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListOperations의 섹션을 참조하세요. AWS CLI

### `list-services`

다음 코드 예시에서는 `list-services`을 사용하는 방법을 보여 줍니다.

AWS CLI

App Runner 서비스의 페이지 매김 목록을 가져오려면

다음 `list-services` 예제에서는 AWS 계정의 모든 App Runner 서비스를 나열합니다. 각 응답에 최대 2개의 서비스가 나열됩니다. 이 예제는 첫 번째 요청을 보여줍니다. 응답에는 두 개의 결과와 다음 요청에 사용할 수 있는 토큰이 포함됩니다. 후속 응답에 토큰이 포함되지 않으면 모든 서비스가 나열됩니다.

```
aws apprunner list-services \
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
    "MaxResults": 2
}
```

출력:

```
{
    "NextToken":
 "eyJDdXN0b21lckFjY291bnRJZCI6IjI3MDIwNTQwMjg0NSIsIlNlcnZpY2VTdGF0dXNDb2RlIjoiUFJPVklTU9OU
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListServices의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

App Runner 서비스와 연결된 태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 App Runner 서비스와 연결된 모든 태그를 나열합니다.

```
aws apprunner list-tags-for-resource \
    --cli-input-json file://input.json
```

`input.json`의 콘텐츠:

```
{
    "ResourceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

출력:

```
{
    "Tags": [
        {
            "Key": "Department",
            "Value": "Retail"
        },
        {
            "Key": "CustomerId",
            "Value": "56439872357912"
        }
    ]
```

```
}
```

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### pause-service

다음 코드 예시에서는 pause-service을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스를 일시 중지하려면

다음 pause-service 예제에서는 App Runner 서비스를 일시 중지합니다.

```
aws apprunner pause-service \
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

출력:

```
{
    "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"
        }
    }
}
```

- 자세한 API 내용은 명령 참조PauseService의 섹션을 참조하세요. AWS CLI

### `resume-service`

다음 코드 예시에서는 `resume-service`을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스를 재개하려면

다음 `resume-service` 예제에서는 App Runner 서비스를 재개합니다.

```
aws apprunner resume-service \
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

출력:

```
{
    "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"
        }
    }
}
```

- 자세한 API 내용은 명령 참조ResumeService의 섹션을 참조하세요. AWS CLI

### `start-deployment`

다음 코드 예시에서는 `start-deployment`을 사용하는 방법을 보여 줍니다.

AWS CLI

수동 배포를 시작하려면

다음 `start-deployment` 예제에서는 App Runner 서비스에 대한 수동 배포를 수행합니다.

```
aws apprunner start-deployment \
    --cli-input-json file://input.json
```

`input.json`의 콘텐츠:

```
{
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa"
}
```

출력:

```
{
```

```
    "OperationId": "853a7d5b-fc9f-4730-831b-fd8037ab832a"
}
```

- 자세한 API 내용은 명령 참조StartDeployment의 섹션을 참조하세요. AWS CLI

### tag-resource

다음 코드 예시에서는 tag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

App Runner 서비스에 태그를 추가하려면

다음 tag-resource 예제에서는 App Runner 서비스에 두 개의 태그를 추가합니다.

```
aws apprunner tag-resource \
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
    "ResourceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
    "Tags": [
        {
            "Key": "Department",
            "Value": "Retail"
        },
        {
            "Key": "CustomerId",
            "Value": "56439872357912"
        }
    ]
}
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### untag-resource

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

App Runner 서비스에서 태그를 제거하려면

다음 `untag-resource` 예제에서는 App Runner 서비스에서 두 개의 태그를 제거합니다.

```
aws apprunner untag-resource \
    --cli-input-json file://input.json
```

`input.json`의 콘텐츠:

```
{
    "ResourceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
    "TagKeys": [
        "Department",
        "CustomerId"
    ]
}
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-service

다음 코드 예시에서는 `update-service`을 사용하는 방법을 보여 줍니다.

AWS CLI

메모리 크기를 업데이트하려면

다음 `update-service` 예제에서는 App Runner 서비스 인스턴스(스케일링 단위)의 메모리 크기를 2048MiB 로 업데이트합니다.

호출이 성공하면 App Runner는 비동기 업데이트 프로세스를 시작합니다. 호출에서 반환되는 `Service` 구조는 이 호출에서 적용 중인 새 메모리 값을 반영합니다.

```
aws apprunner update-service \
    --cli-input-json file://input.json
```

`input.json`의 콘텐츠:

```
{
    "ServiceArn": "arn:aws:apprunner:us-east-1:123456789012:service/python-
app/8fe1e10304f84fd2b0df550fe98a71fa",
    "InstanceConfiguration": {
        "Memory": "4 GB"
    }
}
```

출력:

```
{
    "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"
        }
    }
}
```

- 자세한 API 내용은 명령 참조UpdateService의 섹션을 참조하세요. AWS CLI

## AWS AppConfig 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS AppConfig.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `create-application`

다음 코드 예시에서는 `create-application`을 사용하는 방법을 보여 줍니다.

AWS CLI

애플리케이션을 생성하려면

다음 create-application 예제에서는 에서 애플리케이션을 생성합니다 AWS AppConfig.

```
aws appconfig create-application \
    --name "example-application" \
    --description "An application used for creating an example."
```

출력:

```
{
    "Description": "An application used for creating an example.",
    "Id": "339ohji",
    "Name": "example-application"
}
```

자세한 내용은 AWS AppConfig 사용 설명서의 1단계: AWS AppConfig 애플리케이션 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateApplication의 섹션을 참조하세요. AWS CLI

### create-configuration-profile

다음 코드 예시에서는 create-configuration-profile을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 프로파일을 생성하려면

다음 create-configuration-profile 예제에서는 Systems Manager의 기능인 Parameter Store에 저장된 구성을 사용하여 구성 프로파일을 생성합니다.

```
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"
```

출력:

```
{
    "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
}
```

자세한 내용은 AWS AppConfig 사용 설명서의 3단계: 구성 및 구성 프로필 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateConfigurationProfile의 섹션을 참조하세요. AWS CLI

### `create-environment`

다음 코드 예시에서는 `create-environment`을 사용하는 방법을 보여 줍니다.

AWS CLI

환경을 생성하려면

다음 `create-environment` 예제에서는 create-application을 사용하여 생성한 애플리케이션을 사용하여 Example-Environment라는 AWS AppConfig 환경을 생성합니다.

```
aws appconfig create-environment \
    --application-id "339ohji" \
    --name "Example-Environment"
```

출력:

```
{
    "ApplicationId": "339ohji",
    "Description": null,
    "Id": "54j1r29",
    "Monitors": null,
    "Name": "Example-Environment",
    "State": "ReadyForDeployment"
}
```

자세한 내용은 AWS AppConfig 사용 설명서의 2단계: 환경 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateEnvironment의 섹션을 참조하세요. AWS CLI

### create-extension-association

다음 코드 예시에서는 create-extension-association을 사용하는 방법을 보여 줍니다.

AWS CLI

확장 연결을 생성하려면

다음 create-extension-association 예제에서는 에서 새 확장 연결을 생성합니다 AWS AppConfig.

```
aws appconfig create-extension-association \
    --region us-west-2 \
    --extension-identifier S3-backup-extension \
    --resource-identifier "arn:aws:appconfig:us-west-2:123456789012:application/
Finance" \
    --parameters S3bucket=FinanceConfigurationBackup
```

출력:

```
{
  "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
}
```

자세한 내용은 AWS AppConfig 사용 설명서의 AWS AppConfig 확장 작업을 참조하세요.

- 자세한 API 내용은 명령 참조CreateExtensionAssociation의 섹션을 참조하세요. AWS CLI

### create-extension

다음 코드 예시에서는 create-extension을 사용하는 방법을 보여 줍니다.

AWS CLI

확장을 생성하려면

다음 create-extension 예제에서는 에서 새 확장을 생성합니다 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}
```

출력:

```
{
    "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
        }
    }
}
```

자세한 내용은 AWS AppConfig 사용 설명서의 AWS AppConfig 확장 작업을 참조하세요.

- 자세한 API 내용은 명령 참조CreateExtension의 섹션을 참조하세요. AWS CLI

### create-hosted-configuration-version

다음 코드 예시에서는 create-hosted-configuration-version을 사용하는 방법을 보여 줍니다.

AWS CLI

호스팅 구성 버전을 생성하려면

다음 create-hosted-configuration-version 예제에서는 AWS AppConfig 호스팅 구성 스토어에 새 구성을 생성합니다. 먼저 구성 콘텐츠를 base64로 변환해야 합니다.

```
aws appconfig create-hosted-configuration-version \
    --application-id "339ohji" \
    --configuration-profile-id "ur8hx2f" \
    --
content eyAiTmFtZSI6ICJFeGFtcGxlQXBwbGljYXRpb24iLCAiSWQiOiBFeGFtcGxlSUQsICJSYW5rIjogNyB9
 \
    --content-type "application/json" \
    configuration_version_output_file
```

configuration_version_output_file의 콘텐츠:

```
{ "Name": "ExampleApplication", "Id": ExampleID, "Rank": 7 }
```

출력:

```
{
    "ApplicationId": "339ohji",
    "ConfigurationProfileId": "ur8hx2f",
    "VersionNumber": "1",
    "ContentType": "application/json"
}
```

자세한 내용은 AWS Appconfig 사용 설명서의 AWS AppConfig 호스팅 구성 스토어 정보를 참조하세요.

- 자세한 API 내용은 명령 참조CreateHostedConfigurationVersion의 섹션을 참조하세요. AWS CLI

### delete-application

다음 코드 예시에서는 delete-application을 사용하는 방법을 보여 줍니다.

AWS CLI

애플리케이션 삭제

다음 delete-application 예제에서는 지정된 애플리케이션을 삭제합니다.

```
aws appconfig delete-application \
--application-id 339ohji
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS AppConfig 사용 설명서의 1단계: AWS AppConfig 애플리케이션 생성을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteApplication의 섹션을 참조하세요. AWS CLI

### delete-configuration-profile

다음 코드 예시에서는 delete-configuration-profile을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 프로파일을 삭제하려면

다음 delete-configuration-profile 예제에서는 지정된 구성 프로파일을 삭제합니다.

```
aws appconfig delete-configuration-profile \
    --application-id 339ohji \
    --configuration-profile-id ur8hx2f
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS AppConfig 사용 설명서의 3단계: 구성 및 구성 프로필 생성을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteConfigurationProfile의 섹션을 참조하세요. AWS CLI

### delete-deployment-strategy

다음 코드 예시에서는 delete-deployment-strategy을 사용하는 방법을 보여 줍니다.

AWS CLI

배포 전략을 삭제하려면

다음 delete-deployment-strategy 예제에서는 지정된 배포 전략을 삭제합니다.

```
aws appconfig delete-deployment-strategy \
    --deployment-strategy-id 1225qzk
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS AppConfig 사용 설명서의 4단계: 배포 전략 생성을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteDeploymentStrategy의 섹션을 참조하세요. AWS CLI

### delete-environment

다음 코드 예시에서는 delete-environment을 사용하는 방법을 보여 줍니다.

AWS CLI

환경을 삭제하려면

다음 delete-environment 예제에서는 지정된 애플리케이션 환경을 삭제합니다.

```
aws appconfig delete-environment \
    --application-id 339ohji \
    --environment-id 54j1r29
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS AppConfig 사용 설명서의 2단계: 환경 생성을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteEnvironment의 섹션을 참조하세요. AWS CLI

### delete-extension-association

다음 코드 예시에서는 delete-extension-association을 사용하는 방법을 보여 줍니다.

AWS CLI

확장 연결을 삭제하려면

다음 delete-extension-association 예제에서는 에서 확장 연결을 삭제합니다 AWS AppConfig.

```
aws appconfig delete-extension-association \
```

```
    --region us-west-2 \
    --extension-association-id a1b2c3d4
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS AppConfig 사용 설명서의 AWS AppConfig 확장 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteExtensionAssociation의 섹션을 참조하세요. AWS CLI

### delete-extension

다음 코드 예시에서는 delete-extension을 사용하는 방법을 보여 줍니다.

AWS CLI

확장을 삭제하려면

다음 delete-extension 예제에서는 에서 확장을 삭제합니다 AWS AppConfig.

```
aws appconfig delete-extension \
    --region us-west-2 \
    --extension-identifier S3-backup-extension
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS AppConfig 사용 설명서의 AWS AppConfig 확장 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteExtension의 섹션을 참조하세요. AWS CLI

### delete-hosted-configuration-version

다음 코드 예시에서는 delete-hosted-configuration-version을 사용하는 방법을 보여 줍니다.

AWS CLI

호스팅 구성 버전을 삭제하려면

다음 delete-hosted-configuration-version 예제에서는 호스팅 구성 스토어에서 AWS AppConfig 호스팅되는 구성 버전을 삭제합니다.

```
aws appconfig delete-hosted-configuration-version \
```

```
    --application-id 339ohji \
    --configuration-profile-id ur8hx2f \
    --version-number 1
```

출력:: 이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS AppConfig 사용 설명서의 3단계: 구성 및 구성 프로필 생성을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteHostedConfigurationVersion의 섹션을 참조하세요. AWS CLI

### get-application

다음 코드 예시에서는 get-application을 사용하는 방법을 보여 줍니다.

AWS CLI

애플리케이션의 세부 정보를 나열하려면

다음 get-application 예제에서는 지정된 애플리케이션의 세부 정보를 나열합니다.

```
aws appconfig get-application \
    --application-id 339ohji
```

출력:

```
{
    "Description": "An application used for creating an example.",
    "Id": "339ohji",
    "Name": "example-application"
}
```

자세한 내용은 AWS AppConfig 사용 설명서의 AWS AppConfig 작동 방식을 참조하세요.

- 자세한 API 내용은 명령 참조GetApplication의 섹션을 참조하세요. AWS CLI

### get-configuration-profile

다음 코드 예시에서는 get-configuration-profile을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 프로필 세부 정보를 검색하려면

다음 get-configuration-profile 예제에서는 지정된 구성 프로파일의 세부 정보를 반환합니다.

```
aws appconfig get-configuration-profile \
    --application-id 339ohji \
    --configuration-profile-id ur8hx2f
```

출력:

```
{
    "ApplicationId": "339ohji",
    "Id": "ur8hx2f",
    "Name": "Example-Configuration-Profile",
    "LocationUri": "ssm-parameter://Example-Parameter",
    "RetrievalRoleArn": "arn:aws:iam::111122223333:role/Example-App-Config-Role"
}
```

자세한 내용은 AWS AppConfig 사용 설명서의 3단계: 구성 및 구성 프로필 생성을 참조하세요.

- 자세한 API 내용은 명령 참조GetConfigurationProfile의 섹션을 참조하세요. AWS CLI

### get-configuration

다음 코드 예시에서는 get-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 세부 정보를 검색하려면

다음 get-configuration 예제에서는 예제 애플리케이션의 구성 세부 정보를 반환합니다. 이후 구성 가져오기 호출에서는 버전이 변경된 경우에만 client-configuration-version 파라미터를 사용하여 애플리케이션의 구성을 업데이트합니다. 버전이 변경된 경우에만 구성을 업데이트하면 get-configuration을 호출하여 발생하는 초과 요금이 발생하지 않습니다.

```
aws appconfig get-configuration \
    --application "example-application" \
    --environment "Example-Environment" \
    --configuration "Example-Configuration-Profile" \
    --client-id "test-id" \
    configuration-output-file
```

configuration-output-file의 콘텐츠:

```
{ "Name": "ExampleApplication", "Id": ExampleID, "Rank": 7 }
```

출력:

```
{
    "ConfigurationVersion": "1",
    "ContentType": "application/json"
}
```

자세한 내용은 AWS AppConfig 사용 설명서의 6단계: 구성 수신을 참조하세요.

- 자세한 API 내용은 명령 참조GetConfiguration의 섹션을 참조하세요. AWS CLI

### get-deployment-strategy

다음 코드 예시에서는 get-deployment-strategy을 사용하는 방법을 보여 줍니다.

AWS CLI

배포 전략의 세부 정보를 검색하려면

다음 get-deployment-strategy 예제에서는 지정된 배포 전략의 세부 정보를 나열합니다.

```
aws appconfig get-deployment-strategy \
    --deployment-strategy-id 1225qzk
```

출력:

```
{
    "Id": "1225qzk",
    "Name": "Example-Deployment",
    "DeploymentDurationInMinutes": 15,
    "GrowthType": "LINEAR",
    "GrowthFactor": 25.0,
    "FinalBakeTimeInMinutes": 0,
    "ReplicateTo": "SSM_DOCUMENT"
}
```

자세한 내용은 AWS AppConfig 사용 설명서의 4단계: 배포 전략 생성을 참조하세요.

- 자세한 API 내용은 명령 참조GetDeploymentStrategy의 섹션을 참조하세요. AWS CLI

### get-deployment

다음 코드 예시에서는 get-deployment을 사용하는 방법을 보여 줍니다.

AWS CLI

배포 세부 정보를 검색하려면

다음 get-deployment 예제에서는 지정된 환경 및 배포의 애플리케이션에 대한 배포 세부 정보를 나열합니다.

```
aws appconfig get-deployment \
    --application-id 339ohji \
    --environment-id 54j1r29 \
    --deployment-number 1
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS AppConfig 사용 설명서의 5단계: 구성 배포를 참조하세요.

- 자세한 API 내용은 명령 참조GetDeployment의 섹션을 참조하세요. AWS CLI

### get-environment

다음 코드 예시에서는 get-environment을 사용하는 방법을 보여 줍니다.

AWS CLI

환경 세부 정보를 검색하려면

다음 get-environment 예제에서는 지정된 환경의 세부 정보와 상태를 반환합니다.

```
aws appconfig get-environment \
    --application-id 339ohji \
    --environment-id 54j1r29
```

출력:

```
{
    "ApplicationId": "339ohji",
    "Id": "54j1r29",
    "Name": "Example-Environment",
    "State": "ReadyForDeployment"
}
```

자세한 내용은 AWS AppConfig 사용 설명서의 2단계: 환경 생성을 참조하세요.

- 자세한 API 내용은 명령 참조GetEnvironment의 섹션을 참조하세요. AWS CLI

### get-extension-association

다음 코드 예시에서는 get-extension-association을 사용하는 방법을 보여 줍니다.

AWS CLI

확장 연결 세부 정보를 가져오려면

다음 get-extension-association 예제에서는 확장 연결에 대한 정보를 보여줍니다.

```
aws appconfig get-extension-association \
    --region us-west-2 \
    --extension-association-id a1b2c3d4
```

출력:

```
{
    "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
}
```

자세한 내용은 AWS AppConfig 사용 설명서의 AWS AppConfig 확장 작업을 참조하세요.

- 자세한 API 내용은 명령 참조GetExtensionAssociation의 섹션을 참조하세요. AWS CLI

### get-extension

다음 코드 예시에서는 get-extension을 사용하는 방법을 보여 줍니다.

AWS CLI

확장 세부 정보를 가져오려면

다음 get-extension 예제에서는 확장에 대한 정보를 보여줍니다.

```
aws appconfig get-extension \
    --region us-west-2 \
    --extension-identifier S3-backup-extension
```

출력:

```
{
    "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
        }
    }
}
```

자세한 내용은 AWS AppConfig 사용 설명서의 AWS AppConfig 확장 작업을 참조하세요.

- 자세한 API 내용은 명령 참조GetExtension의 섹션을 참조하세요. AWS CLI

### get-hosted-configuration-version

다음 코드 예시에서는 get-hosted-configuration-version을 사용하는 방법을 보여 줍니다.

AWS CLI

호스팅 구성 세부 정보를 검색하려면

다음 get-hosted-configuration-version 예제에서는 AWS AppConfig 호스팅된 구성의 구성 세부 정보를 검색합니다.

```
aws appconfig get-hosted-configuration-version \
    --application-id 339ohji \
    --configuration-profile-id ur8hx2f \
    --version-number 1 \
    hosted-configuration-version-output
```

hosted-configuration-version-output의 콘텐츠:

```
{ "Name": "ExampleApplication", "Id": ExampleID, "Rank": 7 }
```

출력:

```
{
```

```
    "ApplicationId": "339ohji",
    "ConfigurationProfileId": "ur8hx2f",
    "VersionNumber": "1",
    "ContentType": "application/json"
}
```

자세한 내용은 AWS AppConfig 사용 설명서의 AWS AppConfig 호스팅 구성 스토어 정보를 참조하세요.

- 자세한 API 내용은 명령 참조GetHostedConfigurationVersion의 섹션을 참조하세요. AWS CLI

### `list-applications`

다음 코드 예시에서는 `list-applications`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 애플리케이션을 나열하려면

다음 `list-applications` 예제에서는 AWS 계정에서 사용 가능한 애플리케이션을 나열합니다.

```
aws appconfig list-applications
```

출력:

```
{
    "Items": [
        {
            "Id": "339ohji",
            "Name": "test-application",
            "Description": "An application used for creating an example."
        },
        {
            "Id": "rwalwu7",
            "Name": "Test-Application"
        }
    ]
}
```

자세한 내용은 AWS AppConfig 사용 설명서의 1단계: AWS AppConfig 애플리케이션 생성을 참조하세요.

- 자세한 API 내용은 명령 참조ListApplications의 섹션을 참조하세요. AWS CLI

### `list-configuration-profiles`

다음 코드 예시에서는 `list-configuration-profiles`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 구성 프로파일을 나열하려면

다음 `list-configuration-profiles` 예제에서는 지정된 애플리케이션에 사용할 수 있는 구성 프로파일을 나열합니다.

```
aws appconfig list-configuration-profiles \
    --application-id 339ohji
```

출력:

```
{
    "Items": [
        {
            "ApplicationId": "339ohji",
            "Id": "ur8hx2f",
            "Name": "Example-Configuration-Profile",
            "LocationUri": "ssm-parameter://Example-Parameter"
        }
    ]
}
```

자세한 내용은 AWS AppConfig 사용 설명서의 3단계: 구성 및 구성 프로필 생성을 참조하세요.

- 자세한 API 내용은 명령 참조ListConfigurationProfiles의 섹션을 참조하세요. AWS CLI

### `list-deployment-strategies`

다음 코드 예시에서는 `list-deployment-strategies`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 배포 전략을 나열하려면

다음 `list-deployment-strategies` 예제에서는 AWS 계정에서 사용 가능한 배포 전략을 나열합니다.

```
aws appconfig list-deployment-strategies
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS AppConfig 사용 설명서의 4단계: 배포 전략 생성을 참조하세요.

- 자세한 API 내용은 명령 참조ListDeploymentStrategies의 섹션을 참조하세요. AWS CLI

### `list-deployments`

다음 코드 예시에서는 `list-deployments`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 배포를 나열하려면

다음 `list-deployments` 예제에서는 지정된 애플리케이션 및 환경에 대해 AWS 계정에서 사용 가능한 배포를 나열합니다.

```
aws appconfig list-deployments \
    --application-id 339ohji \
    --environment-id 54j1r29
```

출력:

```
{
    "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"
```

```
        }
    ]
}
```

자세한 내용은 AWS AppConfig 사용 설명서의 5단계: 구성 배포를 참조하세요.

- 자세한 API 내용은 명령 참조ListDeployments의 섹션을 참조하세요. AWS CLI

### `list-environments`

다음 코드 예시에서는 `list-environments`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 환경을 나열하려면

다음 `list-environments` 예제에서는 지정된 애플리케이션에 대해 AWS 계정에서 사용 가능한 환경을 나열합니다.

```
aws appconfig list-environments \
    --application-id 339ohji
```

출력:

```
{
    "Items": [
        {
            "ApplicationId": "339ohji",
            "Id": "54j1r29",
            "Name": "Example-Environment",
            "State": "ReadyForDeployment"
        }
    ]
}
```

자세한 내용은 AWS AppConfig 사용 설명서의 2단계: 환경 생성을 참조하세요.

- 자세한 API 내용은 명령 참조ListEnvironments의 섹션을 참조하세요. AWS CLI

### `list-extension-associations`

다음 코드 예시에서는 `list-extension-associations`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 리전에 대한 AWS 계정의 모든 AWS AppConfig 확장 연결을 나열하려면

다음 `list-extension-associations` 예제에서는 특정 AWS 리전의 현재 AWS 계정에 대한 모든 AWS AppConfig 확장 연결을 나열합니다.

```
aws appconfig list-extension-associations \
    --region us-west-2
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS AppConfig 사용 설명서의 AWS AppConfig 확장 작업을 참조하세요.

- 자세한 API 내용은 명령 참조ListExtensionAssociations의 섹션을 참조하세요. AWS CLI

### `list-extensions`

다음 코드 예시에서는 `list-extensions`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 리전에 대한 AWS 계정의 모든 AWS AppConfig 확장을 나열하려면

다음 `list-extensions` 예제에서는 특정 AWS 리전의 현재 AWS 계정에 대한 모든 AWS AppConfig 확장을 나열합니다. 명령은 사용자 지정 확장과 AWS 작성된 확장을 반환합니다.

```
aws appconfig list-extensions \
    --region us-west-2
```

출력:

```
{
    "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."
        }
    ]
}
```

자세한 내용은 AWS AppConfig 사용 설명서의 AWS AppConfig 확장 작업을 참조하세요.

- 자세한 API 내용은 명령 참조ListExtensions의 섹션을 참조하세요. AWS CLI

### `list-hosted-configuration-versions`

다음 코드 예시에서는 `list-hosted-configuration-versions`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 호스팅 구성 버전을 나열하려면

다음 `list-hosted-configuration-versions` 예제에서는 지정된 애플리케이션 및 구성 프로파일에 대해 호스팅된 구성 스토어에서 AWS AppConfig 호스팅되는 구성 버전을 나열합니다.

```
aws appconfig list-hosted-configuration-versions \
    --application-id 339ohji \
    --configuration-profile-id ur8hx2f
```

출력:

```
{
    "Items": [
        {
            "ApplicationId": "339ohji",
            "ConfigurationProfileId": "ur8hx2f",
            "VersionNumber": 1,
            "ContentType": "application/json"
        }
    ]
}
```

자세한 내용은 AWS AppConfig 사용 설명서의 AWS AppConfig 호스팅 구성 스토어 정보를 참조하세요.

- 자세한 API 내용은 명령 참조ListHostedConfigurationVersions의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

애플리케이션의 태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 지정된 애플리케이션의 태그를 나열합니다.

```
aws appconfig list-tags-for-resource \
    --resource-arn arn:aws:appconfig:us-east-1:682428703967:application/339ohji
```

출력:

```
{
    "Tags": {
        "group1": "1"
    }
}
```

자세한 내용은 AWS AppConfig 사용 설명서의 1단계: AWS AppConfig 애플리케이션 생성을 참조하세요.

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### start-deployment

다음 코드 예시에서는 start-deployment을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 배포를 시작하려면

다음 start-deployment 예제에서는 지정된 환경, 배포 전략 및 구성 프로파일을 사용하여 애플리케이션에 대한 배포를 시작합니다.

```
aws appconfig start-deployment \
    --application-id 339ohji \
    --environment-id 54j1r29 \
    --deployment-strategy-id 1225qzk \
    --configuration-profile-id ur8hx2f \
    --configuration-version 1
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS AppConfig 사용 설명서의 5단계: 구성 배포를 참조하세요.

- 자세한 API 내용은 명령 참조StartDeployment의 섹션을 참조하세요. AWS CLI

### stop-deployment

다음 코드 예시에서는 stop-deployment을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 배포를 중지하려면

다음 stop-deployment 예제에서는 애플리케이션 구성을 지정된 환경에 배포하는 것을 중지합니다.

```
aws appconfig stop-deployment \
    --application-id 339ohji \
    --environment-id 54j1r29 \
    --deployment-number 2
```

출력:

```
{
    "DeploymentNumber": 0,
    "DeploymentDurationInMinutes": 0,
    "GrowthFactor": 0.0,
    "FinalBakeTimeInMinutes": 0,
    "PercentageComplete": 0.0
}
```

자세한 내용은 AWS AppConfig 사용 설명서의 5단계: 구성 배포를 참조하세요.

- 자세한 API 내용은 명령 참조StopDeployment의 섹션을 참조하세요. AWS CLI

### tag-resource

다음 코드 예시에서는 tag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

애플리케이션에 태그를 지정하려면

다음 `tag-resource` 예제에서는 애플리케이션 리소스에 태그를 지정합니다.

```
aws appconfig tag-resource \
    --resource-arn arn:aws:appconfig:us-east-1:682428703967:application/339ohji \
    --tags '{"group1" : "1"}'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS AppConfig 사용 설명서의 1단계: AWS AppConfig 애플리케이션 생성을 참조하세요.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### `untag-resource`

다음 코드 예시에서는 `untag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

애플리케이션에서 태그를 제거하려면

다음 `untag-resource` 예제에서는 지정된 애플리케이션에서 group1 태그를 제거합니다.

```
aws appconfig untag-resource \
    --resource-arn arn:aws:appconfig:us-east-1:111122223333:application/339ohji \
    --tag-keys '["group1"]'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS AppConfig 사용 설명서의 1단계: AWS AppConfig 애플리케이션 생성을 참조하세요.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### `update-application`

다음 코드 예시에서는 `update-application`을 사용하는 방법을 보여 줍니다.

AWS CLI

애플리케이션을 업데이트하려면

다음 update-application 예제에서는 지정된 애플리케이션의 이름을 업데이트합니다.

```
aws appconfig update-application \
    --application-id 339ohji \
    --name "Example-Application"
```

출력:

```
{
    "Id": "339ohji",
    "Name": "Example-Application",
    "Description": "An application used for creating an example."
}
```

자세한 내용은 AWS AppConfig 사용 설명서의 1단계: AWS AppConfig 애플리케이션 생성을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateApplication의 섹션을 참조하세요. AWS CLI

### update-configuration-profile

다음 코드 예시에서는 update-configuration-profile을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 프로파일을 업데이트하려면

다음 update-configuration-profile 예제에서는 지정된 구성 프로파일에 대한 설명을 업데이트합니다.

```
aws appconfig update-configuration-profile \
    --application-id 339ohji \
    --configuration-profile-id ur8hx2f \
    --description "Configuration profile used for examples."
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS AppConfig 사용 설명서의 3단계: 구성 및 구성 프로필 생성을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateConfigurationProfile의 섹션을 참조하세요. AWS CLI

### update-deployment-strategy

다음 코드 예시에서는 update-deployment-strategy을 사용하는 방법을 보여 줍니다.

AWS CLI

배포 전략을 업데이트하려면

다음 update-deployment-strategy 예제에서는 지정된 배포 전략에서 최종 베이킹 시간을 20분으로 업데이트합니다.

```
aws appconfig update-deployment-strategy \
    --deployment-strategy-id 1225qzk \
    --final-bake-time-in-minutes 20
```

출력:

```
{
    "Id": "1225qzk",
    "Name": "Example-Deployment",
    "DeploymentDurationInMinutes": 15,
    "GrowthType": "LINEAR",
    "GrowthFactor": 25.0,
    "FinalBakeTimeInMinutes": 20,
    "ReplicateTo": "SSM_DOCUMENT"
}
```

자세한 내용은 AWS AppConfig 사용 설명서의 4단계: 배포 전략 생성을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateDeploymentStrategy의 섹션을 참조하세요. AWS CLI

### update-environment

다음 코드 예시에서는 update-environment을 사용하는 방법을 보여 줍니다.

AWS CLI

환경을 업데이트하려면

다음 update-environment 예제에서는 환경의 설명을 업데이트합니다.

```
aws appconfig update-environment \
    --application-id 339ohji \
    --environment-id 54j1r29 \
    --description "An environment for examples."
```

출력:

```
{
    "ApplicationId": "339ohji",
    "Id": "54j1r29",
    "Name": "Example-Environment",
    "Description": "An environment for examples.",
    "State": "RolledBack"
}
```

자세한 내용은 AWS AppConfig 사용 설명서의 2단계: 환경 생성을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateEnvironment의 섹션을 참조하세요. AWS CLI

### update-extension-association

다음 코드 예시에서는 update-extension-association을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS AppConfig 확장 연결을 업데이트하려면

다음 update-extension-association 예제에서는 의 확장 연결에 새 파라미터 값을 추가합니다 AWS AppConfig.

```
aws appconfig update-extension-association \
    --region us-west-2 \
    --extension-association-id a1b2c3d4 \
    --parameters S3bucket=FinanceMobileApp
```

출력:

```
{
    "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
}
```

자세한 내용은 AWS AppConfig 사용 설명서의 AWS AppConfig 확장 작업을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateExtensionAssociation의 섹션을 참조하세요. AWS CLI

### update-extension

다음 코드 예시에서는 update-extension을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS AppConfig 확장을 업데이트하려면

다음 update-extension 예제에서는 의 확장에 추가 파라미터 키를 추가합니다 AWS AppConfig.

```
aws appconfig update-extension \
    --region us-west-2 \
    --extension-identifier S3-backup-extension \
    --parameters S3bucket={Required=true},CampaignID={Required=false}
```

출력:

```
{
```

```
    "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
        }
    }
}
```

자세한 내용은 AWS AppConfig 사용 설명서의 AWS AppConfig 확장 작업을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateExtension의 섹션을 참조하세요. AWS CLI

### `validate-configuration`

다음 코드 예시에서는 `validate-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

구성을 검증하려면

다음 `validate-configuration` 예제에서는 구성 프로필의 검사기를 사용하여 구성을 검증합니다.

```
aws appconfig validate-configuration \
    --application-id abc1234 \
    --configuration-profile-id ur8hx2f \
    --configuration-version 1
```

명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS AppConfig 사용 설명서의 3단계: 구성 및 구성 프로필 생성을 참조하세요.

- 자세한 API 내용은 명령 참조ValidateConfiguration의 섹션을 참조하세요. AWS CLI

## 를 사용한 Application Auto Scaling 예제 AWS CLI

다음 코드 예제에서는 Application Auto Scaling과 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `delete-scaling-policy`

다음 코드 예시에서는 `delete-scaling-policy`을 사용하는 방법을 보여 줍니다.

AWS CLI

조정 정책을 삭제하려면

이 예제에서는 기본 클러스터에서 실행되는 Amazon ECS 서비스 웹 앱에 대한 조정 정책을 삭제합니다.

명령:

```
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
```

- 자세한 API 내용은 명령 참조DeleteScalingPolicy의 섹션을 참조하세요. AWS CLI

### delete-scheduled-action

다음 코드 예시에서는 delete-scheduled-action을 사용하는 방법을 보여 줍니다.

AWS CLI

예약된 작업 삭제

follwing delete-scheduled-action 예제는 지정된 Amazon AppStream 2.0 플릿에서 지정된 예약된 작업을 삭제합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 https://docs.aws.amazon.com/autoscaling/application/userguide/application-auto-scaling-scheduled-scaling.htmlApplication Auto Scaling 사용 설명서의 예약된 조정을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteScheduledAction의 섹션을 참조하세요. AWS CLI

### deregister-scalable-target

다음 코드 예시에서는 deregister-scalable-target을 사용하는 방법을 보여 줍니다.

AWS CLI

확장 가능한 대상 등록을 취소하려면

이 예제에서는 기본 클러스터에서 실행 중인 웹 앱이라는 Amazon ECS 서비스에 대한 확장 가능한 대상을 등록 취소합니다.

명령:

```
aws application-autoscaling deregister-scalable-target --service-namespace ecs --
scalable-dimension ecs:service:DesiredCount --resource-id service/default/web-app
```

이 예제에서는 사용자 지정 리소스에 대한 확장 가능 대상의 등록을 취소합니다. custom-resource-id.txt 파일에는 사용자 지정 리소스의 경우 Amazon API Gateway 엔드포인트를 통해 사용자 지정 리소스로 가는 경로인 리소스 ID를 식별하는 문자열이 포함되어 있습니다.

명령:

```
aws application-autoscaling deregister-scalable-target --service-namespace custom-resource --scalable-dimension custom-resource:ResourceType:Property --resource-id file://~/custom-resource-id.txt
```

custom-resource-id.txt 파일의 내용:

```
https://example.execute-api.us-west-2.amazonaws.com/prod/scalableTargetDimensions/1-23456789
```

- 자세한 API 내용은 명령 참조DeregisterScalableTarget의 섹션을 참조하세요. AWS CLI

### describe-scalable-targets

다음 코드 예시에서는 describe-scalable-targets을 사용하는 방법을 보여 줍니다.

AWS CLI

확장 가능한 대상을 설명하려면

다음 describe-scalable-targets 예제에서는 ecs 서비스 네임스페이스의 확장 가능한 대상에 대해 설명합니다.

```
aws application-autoscaling describe-scalable-targets \
    --service-namespace ecs
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS Application Auto Scaling 사용 설명서의 Application Auto Scaling과 함께 사용할 수 있는 서비스를 참조하세요. Auto Scaling

- 자세한 API 내용은 명령 참조DescribeScalableTargets의 섹션을 참조하세요. AWS CLI

### describe-scaling-activities

다음 코드 예시에서는 describe-scaling-activities을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 지정된 Amazon ECS 서비스에 대한 조정 활동을 설명하려면

다음 describe-scaling-activities 예제에서는 default 클러스터에서 실행 중인 web-app 라는 Amazon ECS 서비스에 대한 조정 활동을 설명합니다. 출력은 조정 정책에 의해 시작된 조정 활동을 보여줍니다.

```
aws application-autoscaling describe-scaling-activities \
    --service-namespace ecs \
    --resource-id service/default/web-app
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Application Auto Scaling 사용 설명서의 Application Auto Scaling에 대한 조정 활동을 참조하세요. Auto Scaling

예제 2: 지정된 DynamoDB 테이블에 대한 조정 활동을 설명하려면

다음 `describe-scaling-activities` 예제에서는 라는 DynamoDB 테이블의 크기 조정 활동을 설명합니다`TestTable`. 출력에는 두 가지 다른 예약된 작업으로 시작된 조정 활동이 표시됩니다.

```
aws application-autoscaling describe-scaling-activities \
    --service-namespace dynamodb \
    --resource-id table/TestTable
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Application Auto Scaling 사용 설명서의 Application Auto Scaling에 대한 조정 활동을 참조하세요.  Auto Scaling

- 자세한 API 내용은 명령 참조DescribeScalingActivities의 섹션을 참조하세요. AWS CLI

### describe-scaling-policies

다음 코드 예시에서는 describe-scaling-policies을 사용하는 방법을 보여 줍니다.

AWS CLI

조정 정책을 설명하려면

이 예제 명령은 ECS 서비스 네임스페이스의 조정 정책을 설명합니다.

명령:

```
aws application-autoscaling describe-scaling-policies --service-namespace ecs
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeScalingPolicies의 섹션을 참조하세요. AWS CLI

### `describe-scheduled-actions`

다음 코드 예시에서는 `describe-scheduled-actions`을 사용하는 방법을 보여 줍니다.

AWS CLI

예약된 작업을 설명하려면

다음 describe-scheduled-actions 예제에서는 지정된 서비스 네임스페이스에 대해 예약된 작업에 대한 세부 정보를 표시합니다.

```
aws application-autoscaling describe-scheduled-actions \
    --service-namespace dynamodb
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 https://docs.aws.amazon.com/autoscaling/application/userguide/application-auto-scaling-scheduled-scaling.htmlApplication Auto Scaling 사용 설명서의 예약된 조정을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeScheduledActions의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

확장 가능한 대상의 태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 에서 지정한 확장 가능 대상에 연결된 태그 키 이름과 값을 나열합니다ARN.

```
aws application-autoscaling list-tags-for-resource \
    --resource-arn arn:aws:application-autoscaling:us-west-2:123456789012:scalable-target/1234abcd56ab78cd901ef1234567890ab123
```

출력:

```
{
    "Tags": {
        "environment": "production"
    }
}
```

자세한 내용은 Application Auto Scaling 사용 설명서의 Application Auto Scaling에 대한 지원 태그 지정을 참조하세요.  Auto Scaling

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### `put-scaling-policy`

다음 코드 예시에서는 `put-scaling-policy`을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: 사전 정의된 지표 사양을 사용하여 대상 추적 조정 정책 적용

다음 put-scaling-policy 예제에서는 사전 정의된 지표 사양의 대상 추적 조정 정책을 기본 클러스터의 웹 앱이라는 Amazon ECS 서비스에 적용합니다. 이 정책은 스케일 아웃 및 스케일 인 쿨다운 기간을 60초로 하여 서비스의 평균 CPU 사용률을 75%로 유지합니다. 출력에는 사용자를 대신하여 생성된 두 CloudWatch 경보의 ARNs 및 이름이 포함됩니다.

```
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
```

이 예제에서는 현재 디렉터리에 다음 내용이 포함된 config.json 파일이 있다고 가정합니다.

```
{
     "TargetValue": 75.0,
     "PredefinedMetricSpecification": {
         "PredefinedMetricType": "ECSServiceAverageCPUUtilization"
     },
     "ScaleOutCooldown": 60,
    "ScaleInCooldown": 60
}
```

출력:

```
{
    "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"
        }
    ]
}
```

예 2: 사용자 지정된 지표 사양을 사용하여 대상 추적 조정 정책 적용

다음 put-scaling-policy 예제에서는 사용자 지정 지표 사양이 있는 대상 추적 조정 정책을 기본 클러스터의 웹 앱이라는 Amazon ECS 서비스에 적용합니다. 이 정책은 스케일 아웃 및 스케일 인 쿨다운 기간을 60초로 하여 서비스의 평균 사용률을 75%로 유지합니다. 출력에는 사용자를 대신하여 생성된 두 CloudWatch 경보의 ARNs 및 이름이 포함됩니다.

```
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
```

이 예제에서는 현재 디렉터리에 다음 내용이 포함된 config.json 파일이 있다고 가정합니다.

```
{
    "TargetValue":75.0,
    "CustomizedMetricSpecification":{
        "MetricName":"MyUtilizationMetric",
        "Namespace":"MyNamespace",
        "Dimensions": [
            {
                "Name":"MyOptionalMetricDimensionName",
                "Value":"MyOptionalMetricDimensionValue"
            }
        ],
        "Statistic":"Average",
        "Unit":"Percent"
    },
    "ScaleOutCooldown": 60,
    "ScaleInCooldown": 60
}
```

출력:

```
{
    "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"
        }
    ]
}
```

예 3: 스케일 아웃을 위한 대상 추적 조정 정책 적용

다음 `put-scaling-policy` 예제에서는 대상 추적 조정 정책을 기본 클러스터web-app에서 라는 Amazon ECS 서비스에 적용합니다. 정책은 Application Load Balancer의 `RequestCountPerTarget` 지표가 임계값을 초과할 때 ECS 서비스를 확장하는 데 사용됩니다. 출력에는 사용자를 대신하여 생성된 CloudWatch 경보의 ARN 및 이름이 포함됩니다.

```
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
```

`config.json`의 콘텐츠:

```
{
     "TargetValue": 1000.0,
     "PredefinedMetricSpecification": {
```

```
          "PredefinedMetricType": "ALBRequestCountPerTarget",
          "ResourceLabel": "app/EC2Co-EcsEl-1TKLTMITMM0EO/f37c06a68c1748aa/
targetgroup/EC2Co-Defau-LDNM7Q3ZH1ZN/6d4ea56ca2d6a18d"
      },
      "ScaleOutCooldown": 60,
     "ScaleInCooldown": 60,
     "DisableScaleIn": true
}
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Application Auto Scaling 사용 설명서의 Application Auto Scaling에 대한 대상 추적 조정 정책을 참조하세요. AWS Auto Scaling

- 자세한 API 내용은 명령 참조PutScalingPolicy의 섹션을 참조하세요. AWS CLI

### `put-scheduled-action`

다음 코드 예시에서는 `put-scheduled-action`을 사용하는 방법을 보여 줍니다.

AWS CLI

DynamoDB 테이블에 예약된 작업을 추가하려면

이 예제에서는 라는 DynamoDB 테이블에 예약된 작업을 추가하여 반복 일정에 따라 TestTable 확장합니다. 지정된 일정(매일 오후 12:15UTC)에 현재 용량이 에 지정된 값보다 낮으면 MinCapacityApplication Auto Scaling은 에 지정된 값으로 스케일 아웃됩니다 MinCapacity.

명령:

```
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
```

자세한 내용은 Application Auto Scaling 사용 설명서의 예약된 조정을 참조하세요.

- 자세한 API 내용은 명령 참조PutScheduledAction의 섹션을 참조하세요. AWS CLI

### register-scalable-target

다음 코드 예시에서는 register-scalable-target을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: ECS 서비스를 확장 가능한 대상으로 등록하는 방법

다음 register-scalable-target 예제에서는 Application Auto Scaling 에 Amazon ECS 서비스를 등록합니다. 또한 키 이름과 environment 값이 포함된 태그를 확장 가능한 대상production에 추가합니다.

```
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
```

출력:

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:us-
west-2:123456789012:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

다른 AWS 서비스 및 사용자 지정 리소스에 대한 예제는 AWS Application Auto Scaling 사용 설명서의 Application Auto Scaling과 함께 사용할 수 있는 서비스의 주제를 참조하세요. Auto Scaling

예제 2: 확장 가능한 대상에 대한 조정 활동을 일시 중지하려면

다음 register-scalable-target 예제에서는 확장 가능한 기존 대상에 대한 조정 활동을 일시 중지합니다.

```
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,ScheduledScalingSuspend
```

출력:

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:us-
west-2:123456789012:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

자세한 내용은 Application Auto Scaling 사용 설명서의 Application Auto Scaling에 대한 조정 일시 중지 및 재개를 참조하세요. Auto Scaling

예제 3: 확장 가능한 대상에 대한 조정 활동을 재개하려면

다음 register-scalable-target 예제에서는 확장 가능한 기존 대상에 대한 조정 활동을 재개합니다.

```
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
```

출력:

```
{
    "ScalableTargetARN": "arn:aws:application-autoscaling:us-
west-2:123456789012:scalable-target/1234abcd56ab78cd901ef1234567890ab123"
}
```

자세한 내용은 Application Auto Scaling 사용 설명서의 Application Auto Scaling에 대한 조정 일시 중지 및 재개를 참조하세요. Auto Scaling

- 자세한 API 내용은 명령 참조RegisterScalableTarget의 섹션을 참조하세요. AWS CLI

### `tag-resource`

다음 코드 예시에서는 `tag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

확장 가능한 대상에 태그를 추가하려면

다음 `tag-resource` 예제에서는 키 이름과 `environment` 값이 포함된 태그를 에서 지정한 확장 가능 대상`production`에 추가합니다ARN.

```
aws application-autoscaling tag-resource \
    --resource-arn arn:aws:application-autoscaling:us-west-2:123456789012:scalable-target/1234abcd56ab78cd901ef1234567890ab123 \
    --tags environment=production
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Application Auto Scaling 사용 설명서의 Application Auto Scaling에 대한 지원 태깅을 참조하세요.  Auto Scaling

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### `untag-resource`

다음 코드 예시에서는 `untag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

확장 가능한 대상에서 태그를 제거하려면

다음 `untag-resource` 예제에서는 에서 지정한 확장 가능 대상`environment`에서 키 이름이 인 태그 페어를 제거합니다ARN.

```
aws application-autoscaling untag-resource \
    --resource-arn arn:aws:application-autoscaling:us-west-2:123456789012:scalable-target/1234abcd56ab78cd901ef1234567890ab123 \
    --tag-keys "environment"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Application Auto Scaling 사용 설명서의 Application Auto Scaling에 대한 지원 태그 지정을 참조하세요.  Auto Scaling

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

## 를 사용한 Application Discovery Service 예제 AWS CLI

다음 코드 예제에서는 Application Discovery Service와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `describe-agents`

다음 코드 예시에서는 `describe-agents`을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 collectionStatus 상태의 에이전트 설명

이 예제 명령은 컬렉션 상태가 “STARTED” 또는 “STOPPED”인 컬렉션 에이전트에 대해 설명합니다.

명령:

```
aws discovery describe-agents --filters
 name="collectionStatus",values="STARTED","STOPPED",condition="EQUALS" --max-
results 3
```

출력:

```
{
       "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeAgents의 섹션을 참조하세요. AWS CLI

### `describe-configurations`

다음 코드 예시에서는 `describe-configurations`을 사용하는 방법을 보여 줍니다.

AWS CLI

선택한 자산 구성 설명

이 예제 명령은 지정된 두 서버의 구성을 설명합니다. 이 작업은 구성 ID에서 자산 유형을 감지합니다. 명령당 한 가지 유형의 자산만 허용됩니다.

명령:

```
aws discovery describe-configurations --configuration-ids "d-server-099385097ef9fbcfb" "d-server-0c4f2dd1fee22c6c1"
```

출력:

```
{
    "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":
 "0.04915364583333333",
```

```
                "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-099385097ef9fbcfb",
                "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"
        }
    ]
}
```

선택한 자산 구성 설명

이 예제 명령은 지정된 두 애플리케이션의 구성을 설명합니다. 이 작업은 구성 ID에서 자산 유형을 감지합니다. 명령당 한 가지 유형의 자산만 허용됩니다.

명령:

```
aws discovery describe-configurations --configuration-ids "d-
application-0ac39bc0e4fad0e42" "d-application-02444a45288013764q"
```

출력:

```
{
    "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"
         }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeConfigurations의 섹션을 참조하세요. AWS CLI

### `list-configurations`

다음 코드 예시에서는 `list-configurations`을 사용하는 방법을 보여 줍니다.

AWS CLI

필터 조건 집합을 충족하는 검색된 모든 서버를 나열하려면

이 예제 명령은 두 호스트 이름 패턴 중 하나와 일치하고 Ubuntu를 실행하지 않는 검색된 서버를 나열합니다.

명령:

```
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"
```

출력:

```
{
    "configurations": [
      {
            "server.osVersion": "3.14.48-33.39.amzn1.x86_64",
            "server.type": "EC2",
            "server.hostName": "ip-172-31-42-208",
            "server.timeOfCreation": "2016-10-28 23:44:30.0",
            "server.configurationId": "d-server-099385097ef9fbcfb",
```

```
            "server.osName": "Linux - Amazon Linux AMI release 2015.03",
            "server.agentId": "i-c142b99e"
        },
        {
            "server.osVersion": "3.14.48-33.39.amzn1.x86_64",
            "server.type": "EC2",
            "server.hostName": "ip-172-31-35-152",
            "server.timeOfCreation": "2016-10-28 23:44:00.0",
            "server.configurationId": "d-server-0c4f2dd1fee22c6c1",
            "server.osName": "Linux - Amazon Linux AMI release 2015.03",
            "server.agentId": "i-4447bc1b"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListConfigurations의 섹션을 참조하세요. AWS CLI

## AppRegistry 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AppRegistry.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `associate-attribute-group`

다음 코드 예시에서는 `associate-attribute-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

속성 그룹을 연결하려면

다음 associate-attribute-group 예제는 AWS 계정의 특정 속성 그룹을 AWS 계정의 특정 애플리케이션에 연결합니다.

```
aws servicecatalog-appregistry associate-attribute-group \
    --application "ExampleApplication" \
    --attribute-group "ExampleAttributeGroup"
```

출력:

```
{
    "applicationArn": "arn:aws:servicecatalog:us-west-2:813737243517:/
applications/0ars38r6btoohvpvd9gqrptt9l",
    "attributeGroupArn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-
groups/01sj5xdwhbw54kejwnt09fnpcl"
}
```

자세한 내용은 AWS 서비스 카탈로그 AppRegistry 관리자 안내서의 속성 그룹 연결 및 연결 해제를 참조하세요.

- 자세한 API 내용은 명령 참조AssociateAttributeGroup의 섹션을 참조하세요. AWS CLI

### create-application

다음 코드 예시에서는 create-application을 사용하는 방법을 보여 줍니다.

AWS CLI

애플리케이션을 생성하려면

다음 create-application 예제에서는 AWS 계정에 새 애플리케이션을 생성합니다.

```
aws servicecatalog-appregistry create-application \
    --name "ExampleApplication"
```

출력:

```
{
    "application": {
        "id": "0ars38r6btoohvpvd9gqrptt9l",
        "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/
applications/0ars38r6btoohvpvd9gqrptt9l",
```

```
        "name": "ExampleApplication",
        "creationTime": "2023-02-28T21:10:10.820000+00:00",
        "lastUpdateTime": "2023-02-28T21:10:10.820000+00:00",
        "tags": {}
    }
}
```

자세한 내용은 AWS 서비스 카탈로그 AppRegistry 관리자 안내서의 애플리케이션 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateApplication의 섹션을 참조하세요. AWS CLI

### `create-attribute-group`

다음 코드 예시에서는 `create-attribute-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

속성 그룹을 생성하려면

다음 `create-attribute-group` 예제에서는 AWS 계정에 새 속성 그룹을 생성합니다.

```
aws servicecatalog-appregistry create-attribute-group \
    --name "ExampleAttributeGroup" \
    --attributes '{"SomeKey1":"SomeValue1","SomeKey2":"SomeValue2"}'
```

출력:

```
{
    "attributeGroup": {
        "id": "01sj5xdwhbw54kejwnt09fnpcl",
        "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-
groups/01sj5xdwhbw54kejwnt09fnpcl",
        "name": "ExampleAttributeGroup",
        "creationTime": "2023-02-28T20:38:01.389000+00:00",
        "lastUpdateTime": "2023-02-28T20:38:01.389000+00:00",
        "tags": {}
    }
}
```

자세한 내용은 AWS 서비스 카탈로그 AppRegistry 관리자 안내서의 속성 그룹 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateAttributeGroup의 섹션을 참조하세요. AWS CLI

### delete-application

다음 코드 예시에서는 delete-application을 사용하는 방법을 보여 줍니다.

AWS CLI

애플리케이션 삭제

다음 delete-application 예제에서는 AWS 계정의 특정 애플리케이션을 삭제합니다.

```
aws servicecatalog-appregistry delete-application \
    --application "ExampleApplication3"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS 서비스 카탈로그 AppRegistry 관리자 안내서의 애플리케이션 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteApplication의 섹션을 참조하세요. AWS CLI

### delete-attribute-group

다음 코드 예시에서는 delete-attribute-group을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 8: 속성 그룹 삭제

다음 delete-attribute-group 예제에서는 AWS 계정의 특정 속성 그룹을 삭제합니다.

```
aws servicecatalog-appregistry delete-attribute-group \
    --attribute-group "ExampleAttributeGroup3"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS 서비스 카탈로그 AppRegistry 관리자 안내서의 속성 그룹 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteAttributeGroup의 섹션을 참조하세요. AWS CLI

### get-application

다음 코드 예시에서는 get-application을 사용하는 방법을 보여 줍니다.

AWS CLI

애플리케이션을 가져오려면

다음 get-application 예제에서는 AWS 계정의 특정 애플리케이션에 대한 메타데이터 정보를 검색합니다.

```
aws servicecatalog-appregistry get-application \
    --application "ExampleApplication"
```

출력:

```
{
    "id": "0ars38r6btoohvpvd9gqrptt9l",
```

```
    "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/
applications/0ars38r6btoohvpvd9gqrptt9l",
    "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"
        }
    }
}
```

자세한 내용은 AWS 서비스 카탈로그 AppRegistry 관리자 안내서의 애플리케이션 세부 정보 사용을 참조하세요.

- 자세한 API 내용은 명령 참조GetApplication의 섹션을 참조하세요. AWS CLI

### get-attribute-group

다음 코드 예시에서는 get-attribute-group을 사용하는 방법을 보여 줍니다.

AWS CLI

속성 그룹을 가져오려면

다음 get-attribute-group 예제에서는 AWS 계정의 특정 속성 그룹을 검색합니다.

```
aws servicecatalog-appregistry get-attribute-group \
    --attribute-group "ExampleAttributeGroup"
```

출력:

```
{
    "id": "01sj5xdwhbw54kejwnt09fnpcl",
    "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-
groups/01sj5xdwhbw54kejwnt09fnpcl",
```

```
    "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"
    }
}
```

자세한 내용은 AWS 서비스 카탈로그 AppRegistry 관리자 안내서의 속성 그룹에 대한 메타데이터 관리를 참조하세요.

- 자세한 API 내용은 명령 참조GetAttributeGroup의 섹션을 참조하세요. AWS CLI

### `list-applications`

다음 코드 예시에서는 `list-applications`을 사용하는 방법을 보여 줍니다.

AWS CLI

애플리케이션을 나열하려면

다음 `list-applications` 예제에서는 AWS 계정의 모든 애플리케이션 목록을 검색합니다.

```
aws servicecatalog-appregistry list-applications
```

출력:

```
{
    "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": "0ars38r6btoohvpvd9gqrptt9l",
            "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/
applications/0ars38r6btoohvpvd9gqrptt9l",
            "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"
        }
    ]
}
```

자세한 내용은 AWS 서비스 카탈로그 AppRegistry 관리자 안내서의 애플리케이션 세부 정보 보기를 참조하세요.

- 자세한 API 내용은 명령 참조ListApplications의 섹션을 참조하세요. AWS CLI

### `list-associated-attribute-groups`

다음 코드 예시에서는 `list-associated-attribute-groups`을 사용하는 방법을 보여 줍니다.

AWS CLI

연결된 속성 그룹을 나열하려면

다음 `list-associated-attribute-groups` 예제에서는 AWS 계정의 특정 애플리케이션과 연결된 AWS 계정의 모든 속성 그룹 목록을 검색합니다.

```
aws servicecatalog-appregistry list-associated-attribute-groups \
    --application "ExampleApplication"
```

출력:

```
{
    "attributeGroups": [
        "01sj5xdwhbw54kejwnt09fnpcl"
    ]
}
```

자세한 내용은 AWS 서비스 카탈로그 AppRegistry 관리자 안내서의 속성 그룹 연결 및 연결 해제를 참조하세요.

- 자세한 API 내용은 명령 참조ListAssociatedAttributeGroups의 섹션을 참조하세요. AWS CLI

### `list-attribute-groups-for-application`

다음 코드 예시에서는 `list-attribute-groups-for-application`을 사용하는 방법을 보여 줍니다.

AWS CLI

애플리케이션의 속성 그룹을 나열하려면

다음 `list-attribute-groups-for-application` 예제에서는 AWS 계정의 특정 애플리케이션과 연결된 AWS 계정의 모든 속성 그룹의 세부 정보를 나열합니다.

```
aws servicecatalog-appregistry list-attribute-groups-for-application \
    --application "ExampleApplication"
```

출력:

```
{
    "attributeGroupsDetails": [
        {
            "id": "01sj5xdwhbw54kejwnt09fnpcl",
            "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-
groups/01sj5xdwhbw54kejwnt09fnpcl",
            "name": "ExampleAttributeGroup"
        }
    ]
}
```

자세한 내용은 AWS Service Catalog AppRegistry 관리자 안내서의 속성 그룹 세부 정보 보기를 참조하세요.

- 자세한 API 내용은 명령 참조ListAttributeGroupsForApplication의 섹션을 참조하세요. AWS CLI

### `list-attribute-groups`

다음 코드 예시에서는 `list-attribute-groups`을 사용하는 방법을 보여 줍니다.

AWS CLI

속성 그룹을 나열하려면

다음 `list-attribute-groups` 예제에서는 AWS 계정의 모든 속성 그룹 목록을 검색합니다.

```
aws servicecatalog-appregistry list-attribute-groups
```

출력:

```
{
    "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": "01sj5xdwhbw54kejwnt09fnpcl",
            "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-
groups/01sj5xdwhbw54kejwnt09fnpcl",
            "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"
        }
    ]
}
```

자세한 내용은 AWS Service Catalog AppRegistry 관리자 안내서의 속성 그룹 세부 정보 보기를 참조하세요.

- 자세한 API 내용은 명령 참조ListAttributeGroups의 섹션을 참조하세요. AWS CLI

### update-application

다음 코드 예시에서는 update-application을 사용하는 방법을 보여 줍니다.

AWS CLI

애플리케이션을 업데이트하려면

다음 update-application 예제에서는 AWS 계정의 특정 애플리케이션을 업데이트하여 설명을 포함합니다.

```
aws servicecatalog-appregistry update-application \
    --application "ExampleApplication" \
    --description "This is an example application"
```

출력:

```
{
    "application": {
        "id": "0ars38r6btoohvpvd9gqrptt9l",
        "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/
applications/0ars38r6btoohvpvd9gqrptt9l",
        "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"
        }
    }
}
```

자세한 내용은 AWS 서비스 카탈로그 AppRegistry 관리자 안내서의 애플리케이션 편집을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateApplication의 섹션을 참조하세요. AWS CLI

### update-attribute-group

다음 코드 예시에서는 update-attribute-group을 사용하는 방법을 보여 줍니다.

AWS CLI

속성 그룹을 업데이트하려면

다음 update-attribute-group 예제에서는 AWS 계정의 특정 속성 그룹을 설명으로 업데이트합니다.

```
aws servicecatalog-appregistry update-attribute-group \
    --attribute-group "ExampleAttributeGroup" \
    --description "This is an example attribute group"
```

출력:

```
{
    "attributeGroup": {
        "id": "01sj5xdwhbw54kejwnt09fnpcl",
        "arn": "arn:aws:servicecatalog:us-west-2:813737243517:/attribute-
groups/01sj5xdwhbw54kejwnt09fnpcl",
        "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"
        }
    }
}
```

자세한 내용은 AWS Service Catalog AppRegistry 관리자 안내서의 속성 그룹 편집을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateAttributeGroup의 섹션을 참조하세요. AWS CLI

## 를 사용한 Athena 예제 AWS CLI

다음 코드 예제에서는 Athena와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### batch-get-named-query

다음 코드 예시에서는 batch-get-named-query을 사용하는 방법을 보여 줍니다.

AWS CLI

둘 이상의 쿼리에 대한 정보를 반환하려면

다음 batch-get-named-query 예제에서는 지정된 가 있는 명명된 쿼리에 대한 정보를 반환합니다IDs.

```
aws athena batch-get-named-query \
    --named-query-ids a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 a1b2c3d4-5678-90ab-cdef-EXAMPLE22222 a1b2c3d4-5678-90ab-cdef-EXAMPLE33333
```

출력:

```
{
    "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": []
}
```

자세한 내용은 Amazon Athena 사용 설명서의 Amazon Athena를 사용하여 SQL 쿼리 실행을 참조하세요. Amazon Athena

- 자세한 API 내용은 명령 참조BatchGetNamedQuery의 섹션을 참조하세요. AWS CLI

### batch-get-query-execution

다음 코드 예시에서는 batch-get-query-execution을 사용하는 방법을 보여 줍니다.

AWS CLI

하나 이상의 쿼리 실행에 대한 정보를 반환하려면

다음 batch-get-query-execution 예제에서는 지정된 쿼리 가 있는 쿼리에 대한 쿼리 실행 정보를 반환합니다IDs.

```
aws athena batch-get-query-execution \
    --query-execution-ids a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 a1b2c3d4-5678-90ab-
cdef-EXAMPLE22222
```

출력:

```
{
    "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 'SFO%'
 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": []
}
```

자세한 내용은 Amazon Athena 사용 설명서의 Amazon Athena를 사용하여 SQL 쿼리 실행을 참조하세요. Amazon Athena

- 자세한 API 내용은 명령 참조BatchGetQueryExecution의 섹션을 참조하세요. AWS CLI

### create-data-catalog

다음 코드 예시에서는 create-data-catalog을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 카탈로그를 생성하려면

다음 create-data-catalog 예제에서는 dynamo_db_catalog 데이터 카탈로그를 생성합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다. 결과를 보려면 를 사용합니다aws athena get-data-catalog --name dynamo_db_catalog.

자세한 내용은 Amazon Athena 사용 설명서의 카탈로그 등록: create-data-catalog을 참조하세요.

- 자세한 API 내용은 명령 참조CreateDataCatalog의 섹션을 참조하세요. AWS CLI

### create-named-query

다음 코드 예시에서는 create-named-query을 사용하는 방법을 보여 줍니다.

AWS CLI

명명된 쿼리를 생성하려면

다음 create-named-query 예제에서는 2016년 1월에 시애틀에서 뉴욕으로 향하는 항공편에 대해 flights_parquet 테이블을 쿼리하는 AthenaAdmin 작업 그룹에 저장된 쿼리를 생성합니다. 둘 다 출발 및 도착이 10분 이상 지연되었습니다. 테이블의 공항 코드 값은 큰따옴표가 포함된 문자열(예: “SEA”)이므로 백슬래시로 이스케이프되고 작은따옴표로 둘러싸여 있습니다.

```
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
```

출력:

```
{
    "NamedQueryId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

자세한 내용은 Amazon Athena 사용 설명서의 Amazon Athena를 사용하여 SQL 쿼리 실행을 참조하세요. Amazon Athena

- 자세한 API 내용은 명령 참조CreateNamedQuery의 섹션을 참조하세요. AWS CLI

### create-work-group

다음 코드 예시에서는 create-work-group을 사용하는 방법을 보여 줍니다.

AWS CLI

작업 그룹을 생성하려면

다음 create-work-group 예제에서는 쿼리 결과 출력 위치가 Data_Analyst_Group 인 라는 작업 그룹을 생성합니다s3://awsdoc-example-bucket. 명령은 쿼리 결과 출력 위치를 포함하는 클라이언트 구성 설정을 재정의하는 작업 그룹을 생성합니다. 또한 이 명령은 CloudWatch 지표를 활성화하고 세 개의 키-값 태그 페어를 작업 그룹에 추가하여 다른 작업 그룹과 구분합니다. --configuration 인수에는 옵션을 구분하는 쉼표 앞에 공백이 없습니다.

```
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"
```

이 명령은 출력을 생성하지 않습니다. 결과를 보려면 를 사용합니다aws athena get-work-group --work-group Data_Analyst_Group.

자세한 내용은 Amazon Athena 사용 설명서의 작업 그룹 관리를 참조하세요.

- 자세한 API 내용은 명령 참조CreateWorkGroup의 섹션을 참조하세요. AWS CLI

### delete-data-catalog

다음 코드 예시에서는 delete-data-catalog을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 카탈로그를 삭제하려면

다음 delete-data-catalog 예제에서는 UnusedDataCatalog 데이터 카탈로그를 삭제합니다.

```
aws athena delete-data-catalog \
    --name UnusedDataCatalog
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Athena 사용 설명서의 카탈로그 삭제: delete-data-catalog를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteDataCatalog의 섹션을 참조하세요. AWS CLI

### delete-named-query

다음 코드 예시에서는 delete-named-query을 사용하는 방법을 보여 줍니다.

AWS CLI

명명된 쿼리를 삭제하려면

다음 delete-named-query 예제에서는 지정된 ID가 있는 명명된 쿼리를 삭제합니다.

```
aws athena delete-named-query \
    --named-query-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Athena 사용 설명서의 Amazon Athena를 사용하여 SQL 쿼리 실행을 참조하세요. Amazon Athena

- 자세한 API 내용은 명령 참조DeleteNamedQuery의 섹션을 참조하세요. AWS CLI

### delete-work-group

다음 코드 예시에서는 delete-work-group을 사용하는 방법을 보여 줍니다.

AWS CLI

작업 그룹을 삭제하려면

다음 delete-work-group 예제에서는 TeamB 작업 그룹을 삭제합니다.

```
aws athena delete-work-group \
    --work-group TeamB
```

이 명령은 출력을 생성하지 않습니다. 삭제를 확인하려면 를 사용합니다aws athena list-work-groups.

자세한 내용은 Amazon Athena 사용 설명서의 작업 그룹 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteWorkGroup의 섹션을 참조하세요. AWS CLI

### get-data-catalog

다음 코드 예시에서는 get-data-catalog을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 카탈로그에 대한 정보를 반환하려면

다음 get-data-catalog 예제에서는 dynamo_db_catalog 데이터 카탈로그에 대한 정보를 반환합니다.

```
aws athena get-data-catalog \
    --name dynamo_db_catalog
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 Amazon Athena 사용 설명서의 카탈로그 세부 정보 표시: get-data-catalog를 참조하세요.

- 자세한 API 내용은 명령 참조GetDataCatalog의 섹션을 참조하세요. AWS CLI

### get-database

다음 코드 예시에서는 get-database을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 카탈로그의 데이터베이스에 대한 정보를 반환하려면

다음 get-database 예제에서는 AwsDataCatalog 데이터 카탈로그의 sampledb 데이터베이스에 대한 정보를 반환합니다.

```
aws athena get-database \
    --catalog-name AwsDataCatalog \
    --database-name sampledb
```

출력:

```
{
    "Database": {
        "Name": "sampledb",
        "Description": "Sample database",
        "Parameters": {
            "CreatedBy": "Athena",
            "EXTERNAL": "TRUE"
        }
    }
}
```

자세한 내용은 Amazon Athena 사용 설명서의 데이터베이스 세부 정보: get-database 표시를 참조하세요.

- 자세한 API 내용은 명령 참조GetDatabase의 섹션을 참조하세요. AWS CLI

### get-named-query

다음 코드 예시에서는 get-named-query을 사용하는 방법을 보여 줍니다.

AWS CLI

명명된 쿼리를 반환하려면

다음 get-named-query 예제에서는 지정된 ID가 있는 쿼리에 대한 정보를 반환합니다.

```
aws athena get-named-query \
    --named-query-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

출력:

```
{
    "NamedQuery": {
        "Name": "CloudFront Logs - SFO",
        "Description": "Shows successful GET request data for SFO",
        "Database": "default",
```

```
        "QueryString": "select date, location, browser, uri, status from
 cloudfront_logs where method = 'GET' and status = 200 and location like 'SFO%'
 limit 10",
        "NamedQueryId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "WorkGroup": "AthenaAdmin"
    }
}
```

자세한 내용은 Amazon Athena 사용 설명서의 Amazon Athena를 사용하여 SQL 쿼리 실행을 참조하세요. Amazon Athena

- 자세한 API 내용은 명령 참조GetNamedQuery의 섹션을 참조하세요. AWS CLI

### get-query-execution

다음 코드 예시에서는 get-query-execution을 사용하는 방법을 보여 줍니다.

AWS CLI

쿼리 실행에 대한 정보를 반환하려면

다음 get-query-execution 예제에서는 지정된 쿼리 ID가 있는 쿼리에 대한 정보를 반환합니다.

```
aws athena get-query-execution \
    --query-execution-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

출력:

```
{
    "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 'SFO%' 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"
    }
}
```

자세한 내용은 Amazon Athena 사용 설명서의 Amazon Athena를 사용하여 SQL 쿼리 실행을 참조하세요. Amazon Athena

- 자세한 API 내용은 명령 참조GetQueryExecution의 섹션을 참조하세요. AWS CLI

### `get-query-results`

다음 코드 예시에서는 `get-query-results`을 사용하는 방법을 보여 줍니다.

AWS CLI

쿼리 결과를 반환하려면

다음 `get-query-results` 예제에서는 지정된 쿼리 ID가 있는 쿼리의 결과를 반환합니다.

```
aws athena get-query-results \
    --query-execution-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

출력:

```
{
    "ResultSet": {
        "Rows": [
            {
                "Data": [
                    {
                        "VarCharValue": "date"
```

```
                    },
                    {
                        "VarCharValue": "location"
                    },
                    {
                        "VarCharValue": "browser"
                    },
                    {
                        "VarCharValue": "uri"
                    },
                    {
                        "VarCharValue": "status"
                    }
                ]
            },
            {
                "Data": [
                    {
                        "VarCharValue": "2014-07-05"
                    },
                    {
                        "VarCharValue": "SFO4"
                    },
                    {
                        "VarCharValue": "Safari"
                    },
                    {
                        "VarCharValue": "/test-image-2.jpeg"
                    },
                    {
                        "VarCharValue": "200"
                    }
                ]
            },
            {
                "Data": [
                    {
                        "VarCharValue": "2014-07-05"
                    },
                    {
                        "VarCharValue": "SFO4"
                    },
                    {
                        "VarCharValue": "Opera"
```

```
                    },
                    {
                        "VarCharValue": "/test-image-2.jpeg"
                    },
                    {
                        "VarCharValue": "200"
                    }
                ]
            },
            {
                "Data": [
                    {
                        "VarCharValue": "2014-07-05"
                    },
                    {
                        "VarCharValue": "SFO4"
                    },
                    {
                        "VarCharValue": "Firefox"
                    },
                    {
                        "VarCharValue": "/test-image-3.jpeg"
                    },
                    {
                        "VarCharValue": "200"
                    }
                ]
            },
            {
                "Data": [
                    {
                        "VarCharValue": "2014-07-05"
                    },
                    {
                        "VarCharValue": "SFO4"
                    },
                    {
                        "VarCharValue": "Lynx"
                    },
                    {
                        "VarCharValue": "/test-image-3.jpeg"
                    },
                    {
                        "VarCharValue": "200"
```

```
                    }
                ]
            },
            {
                "Data": [
                    {
                        "VarCharValue": "2014-07-05"
                    },
                    {
                        "VarCharValue": "SFO4"
                    },
                    {
                        "VarCharValue": "IE"
                    },
                    {
                        "VarCharValue": "/test-image-2.jpeg"
                    },
                    {
                        "VarCharValue": "200"
                    }
                ]
            },
            {
                "Data": [
                    {
                        "VarCharValue": "2014-07-05"
                    },
                    {
                        "VarCharValue": "SFO4"
                    },
                    {
                        "VarCharValue": "Opera"
                    },
                    {
                        "VarCharValue": "/test-image-1.jpeg"
                    },
                    {
                        "VarCharValue": "200"
                    }
                ]
            },
            {
                "Data": [
                    {
```

```
                        "VarCharValue": "2014-07-05"
                    },
                    {
                        "VarCharValue": "SFO4"
                    },
                    {
                        "VarCharValue": "Chrome"
                    },
                    {
                        "VarCharValue": "/test-image-3.jpeg"
                    },
                    {
                        "VarCharValue": "200"
                    }
                ]
            },
            {
                "Data": [
                    {
                        "VarCharValue": "2014-07-05"
                    },
                    {
                        "VarCharValue": "SFO4"
                    },
                    {
                        "VarCharValue": "Firefox"
                    },
                    {
                        "VarCharValue": "/test-image-2.jpeg"
                    },
                    {
                        "VarCharValue": "200"
                    }
                ]
            },
            {
                "Data": [
                    {
                        "VarCharValue": "2014-07-05"
                    },
                    {
                        "VarCharValue": "SFO4"
                    },
                    {
```

```
                    "VarCharValue": "Chrome"
                },
                {
                    "VarCharValue": "/test-image-3.jpeg"
                },
                {
                    "VarCharValue": "200"
                }
            ]
        },
        {
            "Data": [
                {
                    "VarCharValue": "2014-07-05"
                },
                {
                    "VarCharValue": "SFO4"
                },
                {
                    "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
}
```

자세한 내용은 Amazon Athena 사용 설명서의 쿼리 결과, 출력 파일 및 쿼리 기록 작업을 참조하세요.

- 자세한 API 내용은 명령 참조GetQueryResults의 섹션을 참조하세요. AWS CLI

### get-table-metadata

다음 코드 예시에서는 get-table-metadata을 사용하는 방법을 보여 줍니다.

AWS CLI

테이블에 대한 메타데이터 정보를 반환하려면

다음 get-table-metadata 예제는 데이터 AwsDataCatalog 카탈로그의 sampledb 데이터베이스에서 열 이름 및 해당 데이터 유형을 포함하여 counties 테이블에 대한 메타데이터 정보를 반환합니다.

```
aws athena get-table-metadata \
    --catalog-name AwsDataCatalog \
    --database-name sampledb \
    --table-name counties
```

출력:

```
{
    "TableMetadata": {
        "Name": "counties",
        "CreateTime": 1593559968.0,
        "LastAccessTime": 0.0,
        "TableType": "EXTERNAL_TABLE",
        "Columns": [
```

```
            {
                "Name": "name",
                "Type": "string",
                "Comment": "from deserializer"
            },
            {
                "Name": "boundaryshape",
                "Type": "binary",
                "Comment": "from deserializer"
            },
            {
                "Name": "motto",
                "Type": "string",
                "Comment": "from deserializer"
            },
            {
                "Name": "population",
                "Type": "int",
                "Comment": "from deserializer"
            }
        ],
        "PartitionKeys": [],
        "Parameters": {
            "EXTERNAL": "TRUE",
            "inputformat": "com.esri.json.hadoop.EnclosedJsonInputFormat",
            "location": "s3://awsdoc-example-bucket/json",
            "outputformat":
 "org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat",
            "serde.param.serialization.format": "1",
            "serde.serialization.lib": "com.esri.hadoop.hive.serde.JsonSerde",
            "transient_lastDdlTime": "1593559968"
        }
    }
}
```

자세한 내용은 Amazon Athena 사용 설명서의 테이블 세부 정보 표시: get-table-metadata를 참조하세요.

- 자세한 API 내용은 명령 참조GetTableMetadata의 섹션을 참조하세요. AWS CLI

### `get-work-group`

다음 코드 예시에서는 `get-work-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

작업 그룹에 대한 정보를 반환하려면

다음 get-work-group 예제에서는 AthenaAdmin 작업 그룹에 대한 정보를 반환합니다.

```
aws athena get-work-group \
    --work-group AthenaAdmin
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 Amazon Athena 사용 설명서의 작업 그룹 관리를 참조하세요.

- 자세한 API 내용은 명령 참조GetWorkGroup의 섹션을 참조하세요. AWS CLI

## list-data-catalogs

다음 코드 예시에서는 list-data-catalogs을 사용하는 방법을 보여 줍니다.

AWS CLI

Athena에 등록된 데이터 카탈로그를 나열하려면

다음 list-data-catalogs 예제에서는 Athena에 등록된 데이터 카탈로그를 나열합니다.

```
aws athena list-data-catalogs
```

출력:

```
{
    "DataCatalogsSummary": [
        {
            "CatalogName": "AwsDataCatalog",
            "Type": "GLUE"
        },
        {
            "CatalogName": "cw_logs_catalog",
            "Type": "LAMBDA"
        },
        {
            "CatalogName": "cw_metrics_catalog",
            "Type": "LAMBDA"
        }
    ]
}
```

자세한 내용은 Amazon Athena 사용 설명서의 등록된 카탈로그 나열: list-data-catalogs을 참조하세요.

- 자세한 API 내용은 명령 참조ListDataCatalogs의 섹션을 참조하세요. AWS CLI

### `list-databases`

다음 코드 예시에서는 `list-databases`을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 카탈로그에 데이터베이스를 나열하려면

다음 `list-databases` 예제에서는 `AwsDataCatalog` 데이터 카탈로그의 데이터베이스를 나열합니다.

```
aws athena list-databases \
    --catalog-name AwsDataCatalog
```

출력:

```
{
    "DatabaseList": [
```

```
        {
            "Name": "default"
        },
        {
            "Name": "mydatabase"
        },
        {
            "Name": "newdb"
        },
        {
            "Name": "sampledb",
            "Description": "Sample database",
            "Parameters": {
                "CreatedBy": "Athena",
                "EXTERNAL": "TRUE"
            }
        },
        {
            "Name": "webdata"
        }
    ]
}
```

자세한 내용은 Amazon Athena 사용 설명서의 카탈로그: list-databases에 데이터베이스 나열을 참조하세요.

- 자세한 API 내용은 명령 참조ListDatabases의 섹션을 참조하세요. AWS CLI

### `list-named-queries`

다음 코드 예시에서는 `list-named-queries`을 사용하는 방법을 보여 줍니다.

AWS CLI

작업 그룹에 대한 명명된 쿼리를 나열하려면

다음 `list-named-queries` 예제에서는 `AthenaAdmin` 작업 그룹에 대한 명명된 쿼리를 나열합니다.

```
aws athena list-named-queries \
    --work-group AthenaAdmin
```

출력:

```
{
    "NamedQueryIds": [
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333"
    ]
}
```

자세한 내용은 Amazon Athena 사용 설명서의 Amazon Athena를 사용하여 SQL 쿼리 실행을 참조하세요. Amazon Athena

- 자세한 API 내용은 명령 참조ListNamedQueries의 섹션을 참조하세요. AWS CLI

### `list-query-executions`

다음 코드 예시에서는 `list-query-executions`을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 작업 그룹에서 쿼리IDs의 쿼리를 나열하려면

다음 `list-query-executions` 예제에서는 `AthenaAdmin` 작업 그룹의 쿼리를 최대 10개IDs까지 나열합니다.

```
aws athena list-query-executions \
    --work-group AthenaAdmin \
    --max-items 10
```

출력:

```
{
    "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": "eyJOZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxMH0="
}
```

자세한 내용은 Amazon Athena 사용 설명서의 쿼리 결과, 출력 파일 및 쿼리 기록 작업을 참조하세요.

- 자세한 API 내용은 명령 참조ListQueryExecutions의 섹션을 참조하세요. AWS CLI

### `list-table-metadata`

다음 코드 예시에서는 `list-table-metadata`을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 카탈로그의 지정된 데이터베이스에 있는 테이블의 메타데이터를 나열하려면

다음 `list-table-metadata` 예제에서는 `AwsDataCatalog` 데이터 카탈로그의 `geography` 데이터베이스에 있는 최대 2개의 테이블에 대한 메타데이터 정보를 반환합니다.

```
aws athena list-table-metadata \
    --catalog-name AwsDataCatalog \
    --database-name geography \
    --max-items 2
```

출력:

```
{
    "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": "eyJOZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ=="
}
```

자세한 내용은 Amazon Athena 사용 설명서의 데이터베이스의 모든 테이블에 대한 메타데이터 표시: list-table-metadata를 참조하세요.

- 자세한 API 내용은 명령 참조ListTableMetadata의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 작업 그룹의 태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 Data_Analyst_Group 작업 그룹의 태그를 나열합니다.

```
aws athena list-tags-for-resource \
    --resource-arn arn:aws:athena:us-west-2:111122223333:workgroup/
Data_Analyst_Group
```

출력:

```
{
    "Tags": [
        {
            "Key": "Division",
            "Value": "West"
        },
        {
            "Key": "Team",
            "Value": "Big Data"
        },
        {
            "Key": "Location",
            "Value": "Seattle"
        }
    ]
```

```
}
```

예제 2: 데이터 카탈로그의 태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 dynamo_db_catalog 데이터 카탈로그의 태그를 나열합니다.

```
aws athena list-tags-for-resource \
    --resource-arn arn:aws:athena:us-west-2:111122223333:datacatalog/
dynamo_db_catalog
```

출력:

```
{
    "Tags": [
        {
            "Key": "Division",
            "Value": "Mountain"
        },
        {
            "Key": "Organization",
            "Value": "Retail"
        },
        {
            "Key": "Product_Line",
            "Value": "Shoes"
        },
        {
            "Key": "Location",
            "Value": "Denver"
        }
    ]
}
```

자세한 내용은 Amazon Athena 사용 설명서의 리소스 태그 나열 list-tags-for-resource을 참조하세요.

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### `list-work-groups`

다음 코드 예시에서는 `list-work-groups`을 사용하는 방법을 보여 줍니다.

AWS CLI

작업 그룹을 나열하려면

다음 `list-work-groups` 예제에서는 현재 계정의 작업 그룹을 나열합니다.

```
aws athena list-work-groups
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 Amazon Athena 사용 설명서의 작업 그룹 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ListWorkGroups의 섹션을 참조하세요. AWS CLI

### `start-query-execution`

다음 코드 예시에서는 `start-query-execution`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 지정된 데이터베이스 및 데이터 카탈로그의 지정된 테이블에 있는 작업 그룹에서 쿼리를 실행하려면

다음 `start-query-execution` 예제에서는 `AthenaAdmin` 작업 그룹을 사용하여 `AwsDataCatalog` 데이터 카탈로그의 에 있는 `cloudfront_logs` 테이블에서 쿼리`cflogsdatabase`를 실행합니다.

```
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 'SFO%' limit 10" \
    --work-group "AthenaAdmin" \
    --query-execution-context Database=cflogsdatabase,Catalog=AwsDataCatalog
```

출력:

```
{
"QueryExecutionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

자세한 내용은 Amazon Athena 사용 설명서의 Amazon Athena를 사용하여 SQL 쿼리 실행을 참조하세요. Amazon Athena

예제 2: 지정된 작업 그룹을 사용하여 지정된 데이터 카탈로그에 데이터베이스를 생성하는 쿼리를 실행하려면

다음 `start-query-execution` 예제에서는 `AthenaAdmin` 작업 그룹을 사용하여 기본 데이터 카탈로그 newdb 에서 데이터베이스를 생성합니다`AwsDataCatalog`.

```
aws athena start-query-execution \
    --query-string "create database if not exists newdb" \
    --work-group "AthenaAdmin"
```

출력:

```
{
"QueryExecutionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11112"
}
```

자세한 내용은 Amazon Athena 사용 설명서의 Amazon Athena를 사용하여 SQL 쿼리 실행을 참조하세요. Amazon Athena

예제 3: 지정된 데이터베이스 및 데이터 카탈로그의 테이블에 뷰를 생성하는 쿼리를 실행하려면

다음 start-query-execution 예제에서는 의 cloudfront_logs 테이블에 있는 SELECT 문을 사용하여 뷰를 cflogsdatabase 생성합니다cf10.

```
aws athena start-query-execution \
    --query-string  "CREATE OR REPLACE VIEW cf10 AS SELECT * FROM cloudfront_logs
 limit 10" \
    --query-execution-context Database=cflogsdatabase
```

출력:

```
{
"QueryExecutionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11113"
}
```

자세한 내용은 Amazon Athena 사용 설명서의 Amazon Athena를 사용하여 SQL 쿼리 실행을 참조하세요. Amazon Athena

- 자세한 API 내용은 명령 참조StartQueryExecution의 섹션을 참조하세요. AWS CLI

### stop-query-execution

다음 코드 예시에서는 stop-query-execution을 사용하는 방법을 보여 줍니다.

AWS CLI

실행 중인 쿼리를 중지하려면

다음 stop-query-execution 예제에서는 지정된 쿼리 ID가 있는 쿼리를 중지합니다.

```
aws athena stop-query-execution \
    --query-execution-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Athena 사용 설명서의 Amazon Athena를 사용하여 SQL 쿼리 실행을 참조하세요. Amazon Athena

- 자세한 API 내용은 명령 참조StopQueryExecution의 섹션을 참조하세요. AWS CLI

### tag-resource

다음 코드 예시에서는 tag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 태그를 추가하려면

다음 tag-resource 예제에서는 dynamo_db_catalog 데이터 카탈로그에 세 개의 태그를 추가합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다. 결과를 보려면 를 사용합니다aws athena list-tags-for-resource --resource-arn arn:aws:athena:us-west-2:111122223333:datacatalog/dynamo_db_catalog.

자세한 내용은 Amazon Athena 사용 설명서의 리소스에 태그 추가: 태그-리소스를 참조하세요.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### untag-resource

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에서 태그를 제거하려면

다음 untag-resource 예제에서는 dynamo_db_catalog 데이터 카탈로그 리소스에서 Specialization 및 Focus 키와 관련 값을 제거합니다.

```
aws athena untag-resource \
    --resource-arn arn:aws:athena:us-west-2:111122223333:datacatalog/
dynamo_db_catalog \
```

```
    --tag-keys Specialization Focus
```

이 명령은 출력을 생성하지 않습니다. 결과를 보려면 `list-tags-for-resource` 명령을 사용합니다.

자세한 내용은 Amazon Athena 사용 설명서의 리소스에서 태그 제거: untag-resource를 참조하세요.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-data-catalog

다음 코드 예시에서는 `update-data-catalog`을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 카탈로그를 업데이트하려면

다음 `update-data-catalog` 예제에서는 Lambda 함수와 `cw_logs_catalog` 데이터 카탈로그에 대한 설명을 업데이트합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다. 결과를 보려면 를 사용합니다`aws athena get-data-catalog --name cw_logs_catalog`.

자세한 내용은 Amazon Athena 사용 설명서의 카탈로그 업데이트: update-data-catalog를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateDataCatalog의 섹션을 참조하세요. AWS CLI

### update-work-group

다음 코드 예시에서는 `update-work-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

작업 그룹을 업데이트하려면

다음 update-work-group 예제에서는 Data_Analyst_Group 작업 그룹을 비활성화합니다. 사용자는 비활성화된 작업 그룹에서 쿼리를 실행하거나 생성할 수 없지만 지표, 데이터 사용량 제한 제어, 작업 그룹 설정, 쿼리 기록 및 저장된 쿼리를 계속 볼 수 있습니다.

```
aws athena update-work-group \
    --work-group Data_Analyst_Group \
    --state DISABLED
```

이 명령은 출력을 생성하지 않습니다. 상태 변화를 확인하려면 aws athena get-work-group --work-group Data_Analyst_Group 를 사용하고 출력의 State 속성을 확인합니다.

자세한 내용은 Amazon Athena 사용 설명서의 작업 그룹 관리를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateWorkGroup의 섹션을 참조하세요. AWS CLI

## 를 사용한 Auto Scaling 예제 AWS CLI

다음 코드 예제에서는 Auto Scaling과 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### attach-instances

다음 코드 예시에서는 attach-instances을 사용하는 방법을 보여 줍니다.

AWS CLI

Auto Scaling 그룹에 인스턴스를 연결하려면

이 예제에서는 지정된 인스턴스를 지정된 Auto Scaling 그룹에 연결합니다.

```
aws autoscaling attach-instances \
    --instance-ids i-061c63c5eb45f0416 \
    --auto-scaling-group-name my-asg
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조AttachInstances의 섹션을 참조하세요. AWS CLI

### attach-load-balancer-target-groups

다음 코드 예시에서는 attach-load-balancer-target-groups을 사용하는 방법을 보여 줍니다.

AWS CLI

Auto Scaling 그룹에 대상 그룹을 연결하는 방법

이 예시에서는 지정된 Auto Scaling 그룹에 지정된 대상 그룹을 연결합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 Elastic Load Balancing 및 Amazon Auto Scaling을 참조하세요.  EC2 Auto Scaling

- 자세한 API 내용은 명령 참조AttachLoadBalancerTargetGroups의 섹션을 참조하세요. AWS CLI

### attach-load-balancers

다음 코드 예시에서는 attach-load-balancers을 사용하는 방법을 보여 줍니다.

AWS CLI

Classic Load Balancer를 Auto Scaling 그룹에 연결하려면

이 예제에서는 지정된 Classic Load Balancer를 지정된 Auto Scaling 그룹에 연결합니다.

```
aws autoscaling attach-load-balancers \
```

```
    --load-balancer-names my-load-balancer \
    --auto-scaling-group-name my-asg
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 Elastic Load Balancing 및 Amazon Auto Scaling을 참조하세요.  EC2 Auto Scaling

- 자세한 API 내용은 명령 참조AttachLoadBalancers의 섹션을 참조하세요. AWS CLI

### `cancel-instance-refresh`

다음 코드 예시에서는 `cancel-instance-refresh`을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스 새로 고침을 취소하려면

다음 `cancel-instance-refresh` 예제에서는 지정된 Auto Scaling 그룹에 대해 진행 중인 인스턴스 새로 고침을 취소합니다.

```
aws autoscaling cancel-instance-refresh \
    --auto-scaling-group-name my-asg
```

출력:

```
{
    "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"
}
```

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 인스턴스 새로 고침 취소를 참조하세요.

- 자세한 API 내용은 명령 참조CancelInstanceRefresh의 섹션을 참조하세요. AWS CLI

### `complete-lifecycle-action`

다음 코드 예시에서는 `complete-lifecycle-action`을 사용하는 방법을 보여 줍니다.

AWS CLI

수명 주기 작업을 완료하려면

이 예제에서는 지정된 수명 주기 작업이 완료되어 인스턴스 시작 또는 종료를 완료할 수 있음을 Amazon EC2 Auto Scaling에 알립니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 Amazon Auto Scaling 수명 주기 후크를 참조하세요.  EC2 Auto Scaling

- 자세한 API 내용은 명령 참조CompleteLifecycleAction의 섹션을 참조하세요. AWS CLI

### `create-auto-scaling-group`

다음 코드 예시에서는 `create-auto-scaling-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: Auto Scaling 그룹을 생성하는 방법

다음 `create-auto-scaling-group` 예시에서는 리전 내 여러 가용 영역의 서브넷에 Auto Scaling 그룹을 생성합니다. 지정된 시작 템플릿의 기본 버전으로 인스턴스가 시작됩니다. 참고로 종료 정책, 상태 확인 구성 등 대부분의 다른 설정에는 기본값이 사용됩니다.

```
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"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Auto Scaling 사용 설명서의 Auto Scaling 그룹을 참조하세요.  EC2 Auto Scaling

예 2: Application Load Balancer, Network Load Balancer 또는 Gateway Load Balancer를 연결하는 방법

이 예제ARN에서는 예상 트래픽을 지원하는 로드 밸런서에 대한 대상 그룹의 를 지정합니다. 상태 확인 유형은 Elastic Load Balancing이 인스턴스를 비정상으로 보고하면 Auto Scaling 그룹이 인스턴스를 교체하도록 ELB를 지정합니다. 또한 이 명령은 상태 확인 유예 기간을 600초로 정의합니다. 유예 기간은 새로 시작된 인스턴스의 조기 종료를 방지하는 데 도움이 됩니다.

```
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"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 Elastic Load Balancing 및 Amazon Auto Scaling을 참조하세요. EC2 Auto Scaling

예 3: 배치 그룹을 지정하고 시작 템플릿의 최신 버전을 사용하는 방법

이 예시에서는 단일 가용 영역 내에 있는 배치 그룹에 인스턴스를 시작합니다. 이는 HPC 워크로드가 있는 지연 시간이 짧은 그룹에 유용할 수 있습니다. 또한 이 예시에서는 그룹의 최소 크기, 최대 크기, 원하는 용량을 지정합니다.

```
aws autoscaling create-auto-scaling-group \
    --auto-scaling-group-name my-asg \
    --launch-template LaunchTemplateId=lt-1234567890abcde12,Version='$Latest' \
    --min-size 1 \
    --max-size 5 \
    --desired-capacity 3 \
    --placement-group my-placement-group \
    --vpc-zone-identifier "subnet-6194ea3b"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Linux 인스턴스용 Amazon 사용 설명서의 배치 그룹을 참조하세요. EC2

예 4: 단일 인스턴스 Auto Scaling 그룹을 지정하고 시작 템플릿의 특정 버전을 사용하는 방법

이 예시에서는 최소 및 최대 용량을 1로 설정한 Auto Scaling 그룹을 생성하여 하나의 인스턴스가 실행되도록 합니다. 명령은 또한 기존 의 ID가 ENI 지정된 시작 템플릿의 v1을 지정합니다. eth0ENI에 기존 를 지정하는 시작 템플릿을 사용하는 경우 요청에 서브넷 ID를 지정하지 않고 네트워크 인터페이스와 일치하는 Auto Scaling 그룹의 가용 영역을 지정해야 합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Auto Scaling 사용 설명서의 Auto Scaling 그룹을 참조하세요. EC2 Auto Scaling

예 5: 다른 종료 정책을 지정하는 방법

이 예시에서는 시작 구성을 사용하여 Auto Scaling 그룹을 생성하고 가장 오래된 인스턴스부터 종료하도록 종료 정책을 설정합니다. 이 명령은 또한 키가 `Role`이고 값이 `WebServer`인 태그를 그룹과 해당 인스턴스에 적용합니다.

```
aws autoscaling create-auto-scaling-group \
    --auto-scaling-group-name my-asg \
    --launch-configuration-name my-lc \
    --min-size 1 \
    --max-size 5 \
    --termination-policies "OldestInstance" \
    --tags "ResourceId=my-asg,ResourceType=auto-scaling-
group,Key=Role,Value=WebServer,PropagateAtLaunch=true" \
    --vpc-zone-identifier "subnet-5ea0c127,subnet-6194ea3b,subnet-c934b782"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 Amazon Auto Scaling 종료 정책 작업을 참조하세요. EC2 Auto Scaling

예 6: 시작 수명 주기 후크를 지정하는 방법

이 예시에서는 인스턴스 시작 시 사용자 지정 작업을 지원하는 수명 주기 후크가 있는 Auto Scaling 그룹을 생성합니다.

```
aws autoscaling create-auto-scaling-group \
    --cli-input-json file://~/config.json
```

config.json 파일의 콘텐츠:

```
{
    "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"
    }]
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 Amazon Auto Scaling 수명 주기 후크를 참조하세요.  EC2 Auto Scaling

예 7: 종료 수명 주기 후크를 지정하는 방법

이 예시에서는 인스턴스 종료 시 사용자 지정 작업을 지원하는 수명 주기 후크가 있는 Auto Scaling 그룹을 생성합니다.

```
aws autoscaling create-auto-scaling-group \
```

```
    --cli-input-json file://~/config.json
```

config.json의 콘텐츠:

```
{
    "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"
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 Amazon Auto Scaling 수명 주기 후크를 참조하세요.  EC2 Auto Scaling

예 8: 사용자 지정 종료 정책을 지정하는 방법

이 예제에서는 Amazon Auto Scaling에 규모에 따라 종료해도 안전한 인스턴스를 알려주는 사용자 지정 Lambda 함수 종료 정책을 지정하는 EC2 Auto Scaling 그룹을 생성합니다.

```
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"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 Lambda를 사용하여 사용자 지정 종료 정책 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateAutoScalingGroup의 섹션을 참조하세요. AWS CLI

### `create-launch-configuration`

다음 코드 예시에서는 `create-launch-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 시작 구성을 생성하려면

이 예제에서는 간단한 시작 구성을 생성합니다.

```
aws autoscaling create-launch-configuration \
    --launch-configuration-name my-lc \
    --image-id ami-04d5cc9b88example \
    --instance-type m5.large
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 시작 구성 생성을 참조하세요.

예제 2: 보안 그룹, 키 페어 및 부트래핑 스크립트를 사용하여 시작 구성을 생성하려면

이 예제에서는 보안 그룹, 키 페어 및 사용자 데이터에 포함된 부트래핑 스크립트를 사용하여 시작 구성을 생성합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 시작 구성 생성을 참조하세요.

예제 3: IAM 역할로 시작 구성을 생성하려면

이 예제에서는 IAM 역할의 인스턴스 프로파일 이름으로 시작 구성을 생성합니다.

```
aws autoscaling create-launch-configuration \
    --launch-configuration-name my-lc \
    --image-id ami-04d5cc9b88example \
    --instance-type m5.large \
    --iam-instance-profile my-autoscaling-role
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IAM Amazon Auto Scaling 사용 설명서의 Amazon EC2 인스턴스에서 실행되는 애플리케이션의 역할을 참조하세요. EC2 Auto Scaling

예제 4: 세부 모니터링이 활성화된 상태로 시작 구성을 생성하려면

이 예제에서는 EC2 세부 모니터링이 활성화된 시작 구성을 생성하여 1분 내에 EC2 지표를 CloudWatch 로 전송합니다.

```
aws autoscaling create-launch-configuration \
    --launch-configuration-name my-lc \
    --image-id ami-04d5cc9b88example \
    --instance-type m5.large \
    --instance-monitoring Enabled=true
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Auto Scaling 사용 설명서의 Auto Scaling 인스턴스에 대한 모니터링 구성을 참조하세요. EC2 Auto Scaling

예제 5: 스팟 인스턴스를 시작하는 시작 구성을 생성하려면

이 예제에서는 스팟 인스턴스를 유일한 구매 옵션으로 사용하는 시작 구성을 생성합니다.

```
aws autoscaling create-launch-configuration \
    --launch-configuration-name my-lc \
    --image-id ami-04d5cc9b88example \
    --instance-type m5.large \
    --spot-price "0.50"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 스팟 인스턴스 요청을 참조하세요.

예제 6: EC2 인스턴스를 사용하여 시작 구성을 생성하려면

이 예제에서는 기존 인스턴스의 속성을 기반으로 시작 구성을 생성합니다. 배치 테넌시와 `--placement-tenancy` 및 `--no-associate-public-ip-address` 옵션을 포함하여 퍼블릭 IP 주소를 설정할지 여부를 재정의합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 EC2 인스턴스를 사용하여 시작 구성 생성을 참조하세요.

예제 7: Amazon EBS 볼륨에 대한 블록 디바이스 매핑을 사용하여 시작 구성을 생성하려면

이 예제에서는 디바이스 이름과 EBS gp3 볼륨 크기가 20인 Amazon `/dev/sdh` 볼륨에 대한 블록 디바이스 매핑을 사용하여 시작 구성을 생성합니다.

```
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"}}]'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Auto Scaling 참조EBS의 섹션을 참조하세요.  EC2 Auto Scaling API

JSON형식이 지정된 파라미터 값을 인용하는 구문에 대한 자세한 내용은 AWS 명령줄 인터페이스 사용 설명서의 에서 문자열로 인용 부호 사용을 참조하세요 AWS CLI.

예제 8: 인스턴스 스토어 볼륨에 대한 블록 디바이스 매핑을 사용하여 시작 구성을 생성하려면

이 예제에서는 를 인스턴스 스토어 볼륨ephemeral1으로 사용하여 디바이스 이름이 인 시작 구성을 생성합니다/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"}]'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Auto Scaling 참조BlockDeviceMapping의 섹션을 참조하세요. EC2 Auto Scaling API

JSON형식이 지정된 파라미터 값을 인용하는 구문에 대한 자세한 내용은 AWS 명령줄 인터페이스 사용 설명서의 에서 문자열과 함께 인용 부호 사용을 참조하세요 AWS CLI.

예제 9: 시작 구성을 생성하고 시작 시 블록 디바이스가 연결되지 않도록 하려면

이 예제에서는 의 블록 디바이스 매핑AMI(예: )에 의해 지정된 블록 디바이스를 억제하는 시작 구성을 생성합니다`/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":""}]'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Auto Scaling 참조BlockDeviceMapping의 섹션을 참조하세요. EC2 Auto Scaling API

따옴표 JSON형식의 파라미터 값을 위한 구문에 대한 자세한 내용은 AWS 명령줄 인터페이스 사용 설명서의 에서 문자열로 따옴표 사용을 참조하세요 AWS CLI.

- 자세한 API 내용은 명령 참조CreateLaunchConfiguration의 섹션을 참조하세요. AWS CLI

### `create-or-update-tags`

다음 코드 예시에서는 `create-or-update-tags`을 사용하는 방법을 보여 줍니다.

AWS CLI

Auto Scaling 그룹에 대한 태그를 생성하거나 업데이트하려면

이 예제에서는 지정된 Auto Scaling 그룹에 두 개의 태그를 추가합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Auto Scaling 사용 설명서의 Auto Scaling 그룹 및 인스턴스 태깅을 참조하세요. EC2 Auto Scaling

- 자세한 API 내용은 명령 참조CreateOrUpdateTags의 섹션을 참조하세요. AWS CLI

### `delete-auto-scaling-group`

다음 코드 예시에서는 `delete-auto-scaling-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: 지정된 Auto Scaling 그룹을 삭제하는 방법

이 예시에서는 지정된 Auto Scaling 그룹을 삭제합니다.

```
aws autoscaling delete-auto-scaling-group \
    --auto-scaling-group-name my-asg
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Auto Scaling 사용 설명서의 Auto Scaling 인프라 삭제를 참조하세요. EC2 Auto Scaling

예 2: 지정된 Auto Scaling 그룹을 강제로 삭제하는 방법

그룹의 인스턴스가 종료될 때까지 기다리지 않고 Auto Scaling 그룹을 삭제하려면 `--force-delete` 옵션을 사용하세요.

```
aws autoscaling delete-auto-scaling-group \
    --auto-scaling-group-name my-asg \
    --force-delete
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 Auto Scaling 인프라 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteAutoScalingGroup의 섹션을 참조하세요. AWS CLI

### `delete-launch-configuration`

다음 코드 예시에서는 `delete-launch-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

시작 구성을 삭제하려면

이 예제에서는 지정된 시작 구성을 삭제합니다.

```
aws autoscaling delete-launch-configuration \
    --launch-configuration-name my-launch-config
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 Auto Scaling 인프라 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteLaunchConfiguration의 섹션을 참조하세요. AWS CLI

### `delete-lifecycle-hook`

다음 코드 예시에서는 `delete-lifecycle-hook`을 사용하는 방법을 보여 줍니다.

AWS CLI

수명 주기 후크를 삭제하려면

이 예제에서는 지정된 수명 주기 후크를 삭제합니다.

```
aws autoscaling delete-lifecycle-hook \
    --lifecycle-hook-name my-lifecycle-hook \
    --auto-scaling-group-name my-asg
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeleteLifecycleHook의 섹션을 참조하세요. AWS CLI

### delete-notification-configuration

다음 코드 예시에서는 delete-notification-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

Auto Scaling 알림을 삭제하려면

이 예제에서는 지정된 Auto Scaling 그룹에서 지정된 알림을 삭제합니다.

```
aws autoscaling delete-notification-configuration \
    --auto-scaling-group-name my-asg \
    --topic-arn arn:aws:sns:us-west-2:123456789012:my-sns-topic
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 알림 구성 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteNotificationConfiguration의 섹션을 참조하세요. AWS CLI

### delete-policy

다음 코드 예시에서는 delete-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

조정 정책을 삭제하려면

이 예제에서는 지정된 조정 정책을 삭제합니다.

```
aws autoscaling delete-policy \
    --auto-scaling-group-name my-asg \
    --policy-name alb1000-target-tracking-scaling-policy
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeletePolicy의 섹션을 참조하세요. AWS CLI

### delete-scheduled-action

다음 코드 예시에서는 delete-scheduled-action을 사용하는 방법을 보여 줍니다.

AWS CLI

Auto Scaling 그룹에서 예약된 작업을 삭제하려면

이 예제에서는 지정된 Auto Scaling 그룹에서 지정된 예약된 작업을 삭제합니다.

```
aws autoscaling delete-scheduled-action \
    --auto-scaling-group-name my-asg \
    --scheduled-action-name my-scheduled-action
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeleteScheduledAction의 섹션을 참조하세요. AWS CLI

## delete-tags

다음 코드 예시에서는 delete-tags을 사용하는 방법을 보여 줍니다.

AWS CLI

Auto Scaling 그룹에서 태그를 삭제하려면

이 예제에서는 지정된 Auto Scaling 그룹에서 지정된 태그를 삭제합니다.

```
aws autoscaling delete-tags \
    --tags ResourceId=my-asg,ResourceType=auto-scaling-group,Key=Dept,Value=Research
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Auto Scaling 사용 설명서의 Auto Scaling 그룹 및 인스턴스 태그를 참조하세요. EC2 Auto Scaling

- 자세한 API 내용은 명령 참조DeleteTags의 섹션을 참조하세요. AWS CLI

## delete-warm-pool

다음 코드 예시에서는 delete-warm-pool을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 웜 풀 삭제

다음 예제에서는 지정된 Auto Scaling 그룹의 웜 풀을 삭제합니다.

```
aws autoscaling delete-warm-pool \
    --auto-scaling-group-name my-asg
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 Amazon Auto Scaling용 웜 풀을 참조하세요. EC2 Auto Scaling

예제 2: 웜 풀을 강제 삭제하려면

인스턴스가 종료될 때까지 기다리지 않고 웜 풀을 삭제하려면 `--force-delete` 옵션을 사용합니다.

```
aws autoscaling delete-warm-pool \
    --auto-scaling-group-name my-asg \
    --force-delete
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 Amazon Auto Scaling용 웜 풀을 참조하세요. EC2 Auto Scaling

- 자세한 API 내용은 명령 참조DeleteWarmPool의 섹션을 참조하세요. AWS CLI

### `describe-account-limits`

다음 코드 예시에서는 `describe-account-limits`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon EC2 Auto Scaling 계정 제한을 설명하려면

이 예제에서는 AWS 계정의 Amazon EC2 Auto Scaling 제한에 대해 설명합니다.

```
aws autoscaling describe-account-limits
```

출력:

```
{
    "NumberOfLaunchConfigurations": 5,
```

```
    "MaxNumberOfLaunchConfigurations": 100,
    "NumberOfAutoScalingGroups": 3,
    "MaxNumberOfAutoScalingGroups": 20
}
```

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 Amazon Auto Scaling 서비스 할당량을 참조하세요. EC2 Auto Scaling

- 자세한 API 내용은 명령 참조DescribeAccountLimits의 섹션을 참조하세요. AWS CLI

### `describe-adjustment-types`

다음 코드 예시에서는 `describe-adjustment-types`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 조정 유형을 설명하려면

이 예제에서는 사용 가능한 조정 유형을 설명합니다.

```
aws autoscaling describe-adjustment-types
```

출력:

```
{
    "AdjustmentTypes": [
        {
            "AdjustmentType": "ChangeInCapacity"
        },
        {
            "AdjustmentType": "ExactCapacity"
        },
        {
            "AdjustmentType": "PercentChangeInCapacity"
        }
    ]
}
```

자세한 내용은 Amazon Auto Scaling 사용 설명서의 조정 유형 조정을 참조하세요. EC2 Auto Scaling

- 자세한 API 내용은 명령 참조DescribeAdjustmentTypes의 섹션을 참조하세요. AWS CLI

### describe-auto-scaling-groups

다음 코드 예시에서는 describe-auto-scaling-groups을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: 지정된 Auto Scaling 그룹을 설명하는 방법

이 예시에서는 지정된 Auto Scaling 그룹을 설명합니다.

```
aws autoscaling describe-auto-scaling-groups \
    --auto-scaling-group-name my-asg
```

출력:

```
{
    "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": []
        }
    ]
}
```

예 2: 처음 100개의 지정된 Auto Scaling 그룹을 설명하는 방법

이 예시에서는 지정된 Auto Scaling 그룹을 설명합니다. 최대 100개의 그룹 이름을 지정할 수 있습니다.

```
aws autoscaling describe-auto-scaling-groups \
    --max-items 100 \
    --auto-scaling-group-name "group1" "group2" "group3" "group4"
```

샘플 출력은 예 1을 참조하세요.

예 3: 지정된 리전에서 Auto Scaling 그룹을 설명하는 방법

이 예시에서는 지정된 리전의 Auto Scaling 그룹을 최대 75개까지 설명합니다.

```
aws autoscaling describe-auto-scaling-groups \
    --max-items 75 \
```

```
    --region us-east-1
```

샘플 출력은 예 1을 참조하세요.

예 4: 지정된 개수의 Auto Scaling 그룹을 설명하는 방법

특정 개수의 Auto Scaling 그룹을 반환하려면 `--max-items` 옵션을 사용하세요.

```
aws autoscaling describe-auto-scaling-groups \
    --max-items 1
```

샘플 출력은 예 1을 참조하세요.

출력에 `NextToken` 필드가 포함된 경우 그룹이 더 많습니다. 추가 그룹을 가져오려면 다음과 같이 후속 직접 호출에서 이 필드의 값을 `--starting-token` 옵션과 함께 사용하세요.

```
aws autoscaling describe-auto-scaling-groups \
    --starting-token Z3M3LMPEXAMPLE
```

샘플 출력은 예 1을 참조하세요.

예제 5: 시작 구성을 사용하는 Auto Scaling 그룹 설명

이 예제에서는 `--query` 옵션을 사용하여 시작 구성을 사용하는 Auto Scaling 그룹을 설명합니다.

```
aws autoscaling describe-auto-scaling-groups \
    --query 'AutoScalingGroups[?LaunchConfigurationName!=`null`]'
```

출력:

```
[
    {
        "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": []
    }
]
```

자세한 내용은 AWS 명령줄 인터페이스 사용 설명서의 출력 필터링 AWS CLI을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeAutoScalingGroups의 섹션을 참조하세요. AWS CLI

### `describe-auto-scaling-instances`

다음 코드 예시에서는 `describe-auto-scaling-instances`을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: 하나 이상의 인스턴스를 설명하는 방법

이 예시에서는 지정된 인스턴스를 설명합니다.

```
aws autoscaling describe-auto-scaling-instances \
    --instance-ids i-06905f55584de02da
```

출력:

```
{
    "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"
            }
        }
    ]
}
```

예 2: 하나 이상의 인스턴스를 설명하는 방법

이 예시에서는 `--max-items` 옵션을 사용하여 이 직접 호출과 함께 반환할 인스턴스 수를 지정합니다.

```
aws autoscaling describe-auto-scaling-instances \
    --max-items 1
```

출력에 `NextToken` 필드가 포함된 경우 인스턴스가 더 많습니다. 추가 인스턴스를 가져오려면 다음과 같이 후속 직접 호출에서 이 필드의 값을 `--starting-token` 옵션과 함께 사용하세요.

```
aws autoscaling describe-auto-scaling-instances \
    --starting-token Z3M3LMPEXAMPLE
```

샘플 출력은 예 1을 참조하세요.

예제 3: 시작 구성을 사용하는 인스턴스 설명

이 예제에서는 `--query` 옵션을 사용하여 시작 구성을 사용하는 인스턴스를 설명합니다.

```
aws autoscaling describe-auto-scaling-instances \
    --query 'AutoScalingInstances[?LaunchConfigurationName!=`null`]'
```

출력:

```
[
    {
        "InstanceId": "i-088c57934a6449037",
        "InstanceType": "t2.micro",
        "AutoScalingGroupName": "my-asg",
        "AvailabilityZone": "us-west-2c",
        "LifecycleState": "InService",
        "HealthStatus": "HEALTHY",
        "LaunchConfigurationName": "my-lc",
        "ProtectedFromScaleIn": false
    }
]
```

자세한 내용은 AWS 명령줄 인터페이스 사용 설명서의 출력 필터링 AWS CLI을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeAutoScalingInstances의 섹션을 참조하세요. AWS CLI

### describe-auto-scaling-notification-types

다음 코드 예시에서는 `describe-auto-scaling-notification-types`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 알림 유형을 설명하려면

이 예제에서는 사용 가능한 알림 유형을 설명합니다.

```
aws autoscaling describe-auto-scaling-notification-types
```

출력:

```
{
    "AutoScalingNotificationTypes": [
        "autoscaling:EC2_INSTANCE_LAUNCH",
        "autoscaling:EC2_INSTANCE_LAUNCH_ERROR",
        "autoscaling:EC2_INSTANCE_TERMINATE",
        "autoscaling:EC2_INSTANCE_TERMINATE_ERROR",
        "autoscaling:TEST_NOTIFICATION"
    ]
}
```

자세한 내용은 Amazon Auto Scaling 사용 설명서의 Auto Scaling 그룹이 확장될 때 Amazon SNS 알림 받기를 참조하세요.  EC2 Auto Scaling

- 자세한 API 내용은 명령 참조DescribeAutoScalingNotificationTypes의 섹션을 참조하세요. AWS CLI

### `describe-instance-refreshes`

다음 코드 예시에서는 `describe-instance-refreshes`을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스 새로 고침을 설명하려면

다음 `describe-instance-refreshes` 예제에서는 상태 메시지 및 상태 이유(사용 가능한 경우)를 포함하여 지정된 Auto Scaling 그룹에 대한 모든 인스턴스 새로 고침 요청에 대한 설명을 반환합니다.

```
aws autoscaling describe-instance-refreshes \
    --auto-scaling-group-name my-asg
```

출력:

```
{
    "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"
            }
        }
    ]
}
```

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 인스턴스 새로 고침 상태 확인을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeInstanceRefreshes의 섹션을 참조하세요. AWS CLI

### `describe-launch-configurations`

다음 코드 예시에서는 `describe-launch-configurations`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 지정된 시작 구성을 설명하려면

이 예제에서는 지정된 시작 구성을 설명합니다.

```
aws autoscaling describe-launch-configurations \
    --launch-configuration-names my-launch-config
```

출력:

```
{
    "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"
            }
        }
    ]
}
```

예제 2: 지정된 수의 시작 구성을 설명하려면

특정 수의 시작 구성을 반환하려면 `--max-items` 옵션을 사용합니다.

```
aws autoscaling describe-launch-configurations \
    --max-items 1
```

출력에 `NextToken` 필드가 포함된 경우 시작 구성이 더 많습니다. 추가 시작 구성을 가져오려면 다음과 같이 후속 통화에서 `--starting-token` 옵션과 함께 이 필드의 값을 사용합니다.

```
aws autoscaling describe-launch-configurations \
    --starting-token Z3M3LMPEXAMPLE
```

- 자세한 API 내용은 명령 참조DescribeLaunchConfigurations의 섹션을 참조하세요. AWS CLI

### describe-lifecycle-hook-types

다음 코드 예시에서는 `describe-lifecycle-hook-types`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 수명 주기 후크 유형을 설명하려면

이 예제에서는 사용 가능한 수명 주기 후크 유형을 설명합니다.

```
aws autoscaling describe-lifecycle-hook-types
```

출력:

```
{
    "LifecycleHookTypes": [
        "autoscaling:EC2_INSTANCE_LAUNCHING",
        "autoscaling:EC2_INSTANCE_TERMINATING"
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeLifecycleHookTypes의 섹션을 참조하세요. AWS CLI

### `describe-lifecycle-hooks`

다음 코드 예시에서는 `describe-lifecycle-hooks`을 사용하는 방법을 보여 줍니다.

AWS CLI

수명 주기 후크를 설명하려면

이 예제에서는 지정된 Auto Scaling 그룹의 수명 주기 후크를 설명합니다.

```
aws autoscaling describe-lifecycle-hooks \
    --auto-scaling-group-name my-asg
```

출력:

```
{
    "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"
        }
```

```
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeLifecycleHooks의 섹션을 참조하세요. AWS CLI

### `describe-load-balancer-target-groups`

다음 코드 예시에서는 `describe-load-balancer-target-groups`을 사용하는 방법을 보여 줍니다.

AWS CLI

Auto Scaling 그룹의 로드 밸런서 대상 그룹을 설명하려면

이 예제에서는 지정된 Auto Scaling 그룹에 연결된 로드 밸런서 대상 그룹에 대해 설명합니다.

```
aws autoscaling describe-load-balancer-target-groups \
    --auto-scaling-group-name my-asg
```

출력:

```
{
    "LoadBalancerTargetGroups": [
        {
            "LoadBalancerTargetGroupARN": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067",
            "State": "Added"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeLoadBalancerTargetGroups의 섹션을 참조하세요. AWS CLI

### `describe-load-balancers`

다음 코드 예시에서는 `describe-load-balancers`을 사용하는 방법을 보여 줍니다.

AWS CLI

Auto Scaling 그룹의 Classic Load Balancer를 설명하려면

이 예제에서는 지정된 Auto Scaling 그룹의 Classic Load Balancer에 대해 설명합니다.

```
aws autoscaling describe-load-balancers \
    --auto-scaling-group-name my-asg
```

출력:

```
{
    "LoadBalancers": [
        {
            "State": "Added",
            "LoadBalancerName": "my-load-balancer"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeLoadBalancers의 섹션을 참조하세요. AWS CLI

### describe-metric-collection-types

다음 코드 예시에서는 describe-metric-collection-types을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 지표 컬렉션 유형을 설명하려면

이 예제에서는 사용 가능한 지표 컬렉션 유형을 설명합니다.

```
aws autoscaling describe-metric-collection-types
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Auto Scaling 사용 설명서의 Auto Scaling 그룹 지표를 참조하세요. EC2 Auto Scaling

- 자세한 API 내용은 명령 참조DescribeMetricCollectionTypes의 섹션을 참조하세요. AWS CLI

### describe-notification-configurations

다음 코드 예시에서는 describe-notification-configurations을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 지정된 그룹의 알림 구성을 설명하려면

이 예제에서는 지정된 Auto Scaling 그룹에 대한 알림 구성을 설명합니다.

```
aws autoscaling describe-notification-configurations \
    --auto-scaling-group-name my-asg
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Auto Scaling 사용 설명서의 Auto Scaling 그룹이 확장될 때 Amazon SNS 알림 받기를 참조하세요. EC2 Auto Scaling

예제 1: 지정된 수의 알림 구성을 설명하려면

특정 수의 알림 구성을 반환하려면 max-items 파라미터를 사용합니다.

```
aws autoscaling describe-notification-configurations \
    --auto-scaling-group-name my-auto-scaling-group \
    --max-items 1
```

출력:

```
{
    "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"
        }
    ]
}
```

출력에 NextToken 필드가 포함된 경우 알림 구성이 더 많습니다. 추가 알림 구성을 가져오려면 다음과 같이 후속 호출에서 starting-token 파라미터와 함께 이 필드의 값을 사용합니다.

```
aws autoscaling describe-notification-configurations \
    --auto-scaling-group-name my-asg \
    --starting-token Z3M3LMPEXAMPLE
```

자세한 내용은 Amazon Auto Scaling 사용 설명서의 Auto Scaling 그룹이 확장될 때 Amazon SNS 알림 받기를 참조하세요.  EC2 Auto Scaling

- 자세한 API 내용은 명령 참조DescribeNotificationConfigurations의 섹션을 참조하세요. AWS CLI

### describe-policies

다음 코드 예시에서는 describe-policies을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 지정된 그룹의 조정 정책을 설명하려면

이 예제에서는 지정된 Auto Scaling 그룹에 대한 조정 정책을 설명합니다.

```
aws autoscaling describe-policies \
    --auto-scaling-group-name my-asg
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 Amazon Auto Scaling 사용 설명서의 동적 조정을 참조하세요.  EC2 Auto Scaling

예제 2: 지정된 이름의 조정 정책을 설명하는 방법

특정 조정 정책을 반환하려면 `--policy-names` 옵션을 사용합니다.

```
aws autoscaling describe-policies \
    --auto-scaling-group-name my-asg \
    --policy-names cpu40-target-tracking-scaling-policy
```

샘플 출력은 예 1을 참조하세요.

자세한 내용은 Amazon Auto Scaling 사용 설명서의 동적 조정을 참조하세요. EC2 Auto Scaling

예제 3: 여러 조정 정책을 설명하는 방법

특정 수의 정책을 반환하려면 `--max-items` 옵션을 사용합니다.

```
aws autoscaling describe-policies \
    --auto-scaling-group-name my-asg \
    --max-items 1
```

샘플 출력은 예 1을 참조하세요.

출력에 `NextToken` 필드가 포함된 경우 이 필드의 값을 후속 호출의 `--starting-token` 옵션과 함께 사용하여 추가 정책을 가져옵니다.

```
aws autoscaling describe-policies --auto-scaling-group-name my-asg --starting-
token Z3M3LMPEXAMPLE
```

자세한 내용은 Amazon Auto Scaling 사용 설명서의 동적 조정을 참조하세요. EC2 Auto Scaling

- 자세한 API 내용은 명령 참조DescribePolicies의 섹션을 참조하세요. AWS CLI

### `describe-scaling-activities`

다음 코드 예시에서는 `describe-scaling-activities`을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: 지정된 그룹에 대한 크기 조정 활동을 설명하는 방법

이 예시에서는 지정된 Auto Scaling 그룹에 대한 크기 조정 활동을 설명합니다.

```
aws autoscaling describe-scaling-activities \
    --auto-scaling-group-name my-asg
```

출력:

```
{
    "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\"}"
        }
    ]
}
```

자세한 내용은 Amazon Auto Scaling 사용 설명서의 Auto Scaling 그룹에 대한 조정 활동 확인을 참조하세요. EC2 Auto Scaling

예 2: 삭제된 그룹에 대한 크기 조정 활동을 설명하는 방법

Auto Scaling 그룹이 삭제된 후 크기 조정 활동을 설명하려면 `--include-deleted-groups` 옵션을 추가하세요.

```
aws autoscaling describe-scaling-activities \
    --auto-scaling-group-name my-asg \
    --include-deleted-groups
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 Amazon Auto Scaling 문제 해결을 참조하세요. EC2 Auto Scaling

예 3: 지정된 개수의 크기 조정 활동을 설명하는 방법

특정 개수의 활동을 반환하려면 `--max-items` 옵션을 사용하세요.

```
aws autoscaling describe-scaling-activities \
    --max-items 1
```

출력:

```
{
    "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\"}"
        }
    ]
}
```

출력에 NextToken 필드가 포함된 경우 활동이 더 많습니다. 추가 활동을 가져오려면 다음과 같이 후속 직접 호출에서 이 필드의 값을 --starting-token 옵션과 함께 사용하세요.

```
aws autoscaling describe-scaling-activities \
    --starting-token Z3M3LMPEXAMPLE
```

자세한 내용은 Amazon Auto Scaling 사용 설명서의 Auto Scaling 그룹에 대한 조정 활동 확인을 참조하세요. EC2 Auto Scaling

- 자세한 API 내용은 명령 참조DescribeScalingActivities의 섹션을 참조하세요. AWS CLI

### describe-scaling-process-types

다음 코드 예시에서는 describe-scaling-process-types을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 프로세스 유형을 설명하려면

이 예제에서는 사용 가능한 프로세스 유형을 설명합니다.

```
aws autoscaling describe-scaling-process-types
```

출력:

```
{
    "Processes": [
        {
            "ProcessName": "AZRebalance"
        },
        {
            "ProcessName": "AddToLoadBalancer"
```

```
        },
        {
            "ProcessName": "AlarmNotification"
        },
        {
            "ProcessName": "HealthCheck"
        },
        {
            "ProcessName": "InstanceRefresh"
        },
        {
            "ProcessName": "Launch"
        },
        {
            "ProcessName": "ReplaceUnhealthy"
        },
        {
            "ProcessName": "ScheduledActions"
        },
        {
            "ProcessName": "Terminate"
        }
    ]
}
```

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 조정 프로세스 일시 중지 및 재개를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeScalingProcessTypes의 섹션을 참조하세요. AWS CLI

### describe-scheduled-actions

다음 코드 예시에서는 describe-scheduled-actions을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 예약된 모든 작업을 설명하려면

이 예제에서는 예약된 모든 작업에 대해 설명합니다.

```
aws autoscaling describe-scheduled-actions
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 예약된 조정을 참조하세요.

예제 2: 지정된 그룹에 대해 예약된 작업을 설명하려면

특정 Auto Scaling 그룹에 대해 예약된 작업을 설명하려면 `--auto-scaling-group-name` 옵션을 사용합니다.

```
aws autoscaling describe-scheduled-actions \
    --auto-scaling-group-name my-asg
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Auto Scaling 사용 설명서의 예약된 조정을 참조하세요. EC2 Auto Scaling

예제 3: 지정된 예약된 작업을 설명하려면

특정 예약된 작업을 설명하려면 `--scheduled-action-names` 옵션을 사용합니다.

```
aws autoscaling describe-scheduled-actions \
    --scheduled-action-names my-recurring-action
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Auto Scaling 사용 설명서의 예약된 조정을 참조하세요. EC2 Auto Scaling

예제 4: 지정된 시작 시간으로 예약된 작업을 설명하는 방법

특정 시간에 시작하는 예약된 작업을 설명하려면 `--start-time` 옵션을 사용합니다.

```
aws autoscaling describe-scheduled-actions \
    --start-time "2023-12-01T04:00:00Z"
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 예약된 조정을 참조하세요.

예제 5: 지정된 시간에 종료되는 예약된 작업을 설명하려면

특정 시간에 종료되는 예약된 작업을 설명하려면 `--end-time` 옵션을 사용합니다.

```
aws autoscaling describe-scheduled-actions \
    --end-time "2023-12-01T04:00:00Z"
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 예약된 조정을 참조하세요.

예제 6: 지정된 수의 예약된 작업을 설명하려면

특정 수의 예약된 작업을 반환하려면 `--max-items` 옵션을 사용합니다.

```
aws autoscaling describe-scheduled-actions \
    --auto-scaling-group-name my-asg \
    --max-items 1
```

출력:

```
{
    "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"
        }
    ]
}
```

출력에 NextToken 필드가 포함된 경우 더 많은 예약된 작업이 있습니다. 추가 예약된 작업을 가져오려면 다음과 같이 후속 통화에서 --starting-token 옵션과 함께 이 필드의 값을 사용합니다.

```
aws autoscaling describe-scheduled-actions \
    --auto-scaling-group-name my-asg \
    --starting-token Z3M3LMPEXAMPLE
```

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 예약된 조정을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeScheduledActions의 섹션을 참조하세요. AWS CLI

### describe-tags

다음 코드 예시에서는 describe-tags을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 태그를 설명하려면

이 예제에서는 모든 태그를 설명합니다.

```
aws autoscaling describe-tags
```

출력:

```
{
    "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"
        }
    ]
```

```
}
```

자세한 내용은 Amazon Auto Scaling 사용 설명서의 Auto Scaling 그룹 및 인스턴스 태그를 참조하세요. EC2 Auto Scaling

예제 2: 지정된 그룹에 대한 태그를 설명하려면

특정 Auto Scaling 그룹에 대한 태그를 설명하려면 `--filters` 옵션을 사용합니다.

```
aws autoscaling describe-tags --filters Name=auto-scaling-group,Values=my-asg
```

자세한 내용은 Amazon Auto Scaling 사용 설명서의 Auto Scaling 그룹 및 인스턴스 태그를 참조하세요. EC2 Auto Scaling

예제 3: 지정된 태그 수를 설명하려면

특정 수의 태그를 반환하려면 `--max-items` 옵션을 사용합니다.

```
aws autoscaling describe-tags \
    --max-items 1
```

출력에 `NextToken` 필드가 포함된 경우 태그가 더 많습니다. 추가 태그를 가져오려면 다음과 같이 후속 통화에서 `--starting-token` 옵션과 함께 이 필드의 값을 사용합니다.

```
aws autoscaling describe-tags \
    --filters Name=auto-scaling-group,Values=my-asg \
    --starting-token Z3M3LMPEXAMPLE
```

자세한 내용은 Amazon Auto Scaling 사용 설명서의 Auto Scaling 그룹 및 인스턴스 태그 지정을 참조하세요. EC2 Auto Scaling

- 자세한 API 내용은 명령 참조DescribeTags의 섹션을 참조하세요. AWS CLI

### `describe-termination-policy-types`

다음 코드 예시에서는 `describe-termination-policy-types`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 종료 정책 유형을 설명하려면

이 예제에서는 사용 가능한 종료 정책 유형을 설명합니다.

```
aws autoscaling describe-termination-policy-types
```

출력:

```
{
    "TerminationPolicyTypes": [
        "AllocationStrategy",
        "ClosestToNextInstanceHour",
        "Default",
        "NewestInstance",
        "OldestInstance",
        "OldestLaunchConfiguration",
        "OldestLaunchTemplate"
    ]
}
```

자세한 내용은 Amazon Auto Scaling 사용 설명서의 에서 스케일링 중에 종료되는 Auto Scaling 인스턴스 제어를 참조하세요. EC2 Auto Scaling

- 자세한 API 내용은 명령 참조DescribeTerminationPolicyTypes의 섹션을 참조하세요. AWS CLI

### describe-warm-pool

다음 코드 예시에서는 describe-warm-pool을 사용하는 방법을 보여 줍니다.

AWS CLI

웜 풀을 설명하려면

이 예제에서는 지정된 Auto Scaling 그룹의 웜 풀을 설명합니다.

```
aws autoscaling describe-warm-pool \
    --auto-scaling-group-name my-asg
```

출력:

```
{
    "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"
            }
        }
    ]
}
```

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 Amazon Auto Scaling용 웜 풀을 참조하세요.  EC2 Auto Scaling

- 자세한 API 내용은 명령 참조DescribeWarmPool의 섹션을 참조하세요. AWS CLI

### `detach-instances`

다음 코드 예시에서는 `detach-instances`을 사용하는 방법을 보여 줍니다.

AWS CLI

Auto Scaling 그룹에서 인스턴스를 분리하려면

이 예제에서는 지정된 Auto Scaling 그룹에서 지정된 인스턴스를 분리합니다.

```
aws autoscaling detach-instances \
```

```
    --instance-ids i-030017cfa84b20135 \
    --auto-scaling-group-name my-asg \
    --should-decrement-desired-capacity
```

출력:

```
{
    "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\"}"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DetachInstances의 섹션을 참조하세요. AWS CLI

### `detach-load-balancer-target-groups`

다음 코드 예시에서는 `detach-load-balancer-target-groups`을 사용하는 방법을 보여 줍니다.

AWS CLI

Auto Scaling 그룹에서 로드 밸런서 대상 그룹을 분리하려면

이 예제에서는 지정된 Auto Scaling 그룹에서 지정된 로드 밸런서 대상 그룹을 분리합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Auto Scaling 사용 설명서의 Auto Scaling 그룹에 로드 밸런서 연결을 참조하세요. EC2 Auto Scaling

- 자세한 API 내용은 명령 참조DetachLoadBalancerTargetGroups의 섹션을 참조하세요. AWS CLI

### `detach-load-balancers`

다음 코드 예시에서는 `detach-load-balancers`을 사용하는 방법을 보여 줍니다.

AWS CLI

Auto Scaling 그룹에서 Classic Load Balancer를 분리하려면

이 예제에서는 지정된 Auto Scaling 그룹에서 지정된 Classic Load Balancer를 분리합니다.

```
aws autoscaling detach-load-balancers \
    --load-balancer-names my-load-balancer \
    --auto-scaling-group-name my-asg
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Auto Scaling 사용 설명서의 Auto Scaling 그룹에 로드 밸런서 연결을 참조하세요. EC2 Auto Scaling

- 자세한 API 내용은 명령 참조DetachLoadBalancers의 섹션을 참조하세요. AWS CLI

### `disable-metrics-collection`

다음 코드 예시에서는 `disable-metrics-collection`을 사용하는 방법을 보여 줍니다.

AWS CLI

Auto Scaling 그룹 지표 수집을 비활성화는 방법

이 예시에서는 지정된 Auto Scaling 그룹에 대한 `GroupDesiredCapacity` 지표 수집을 비활성화합니다.

```
aws autoscaling disable-metrics-collection \
    --auto-scaling-group-name my-asg \
    --metrics GroupDesiredCapacity
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Auto Scaling 사용 설명서의 Auto Scaling 그룹 및 인스턴스에 대한 CloudWatch 지표 모니터링을 참조하세요. EC2 Auto Scaling

- 자세한 API 내용은 명령 참조DisableMetricsCollection의 섹션을 참조하세요. AWS CLI

### enable-metrics-collection

다음 코드 예시에서는 enable-metrics-collection을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: Auto Scaling 그룹 지표 수집을 활성화하는 방법

이 예시에서는 지정된 Auto Scaling 그룹에 대한 데이터 수집을 활성화합니다.

```
aws autoscaling enable-metrics-collection \
    --auto-scaling-group-name my-asg \
    --granularity "1Minute"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Auto Scaling 사용 설명서의 Auto Scaling 그룹 및 인스턴스에 대한 CloudWatch 지표 모니터링을 참조하세요. EC2 Auto Scaling

예 2: Auto Scaling 그룹의 지정된 지표에 대한 데이터를 수집하는 방법

특정 지표에 대한 데이터를 수집하려면 --metrics 옵션을 사용하세요.

```
aws autoscaling enable-metrics-collection \
    --auto-scaling-group-name my-asg \
    --metrics GroupDesiredCapacity --granularity "1Minute"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Auto Scaling 사용 설명서의 Auto Scaling 그룹 및 인스턴스에 대한 CloudWatch 지표 모니터링을 참조하세요. EC2 Auto Scaling

- 자세한 API 내용은 명령 참조EnableMetricsCollection의 섹션을 참조하세요. AWS CLI

### enter-standby

다음 코드 예시에서는 enter-standby을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스를 대기 모드로 이동하려면

이 예제에서는 지정된 인스턴스를 대기 모드로 전환합니다. 이는 현재 사용 중인 인스턴스를 업데이트하거나 문제를 해결하는 데 유용합니다.

```
aws autoscaling enter-standby \
    --instance-ids i-061c63c5eb45f0416 \
    --auto-scaling-group-name my-asg \
    --should-decrement-desired-capacity
```

출력:

```
{
    "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\"}"
        }
    ]
}
```

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 Amazon Auto Scaling 인스턴스 수명 주기를 참조하세요.  EC2 Auto Scaling

- 자세한 API 내용은 명령 참조EnterStandby의 섹션을 참조하세요. AWS CLI

### `execute-policy`

다음 코드 예시에서는 `execute-policy`을 사용하는 방법을 보여 줍니다.

AWS CLI

조정 정책을 실행하려면

이 예제에서는 지정된 Auto Scaling 그룹에 `my-step-scale-out-policy` 대해 라는 조정 정책을 실행합니다.

```
aws autoscaling execute-policy \
    --auto-scaling-group-name my-asg \
    --policy-name  my-step-scale-out-policy \
    --metric-value 95 \
    --breach-threshold 80
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 단계 및 간단한 조정 정책을 참조하세요.

- 자세한 API 내용은 명령 참조ExecutePolicy의 섹션을 참조하세요. AWS CLI

### `exit-standby`

다음 코드 예시에서는 `exit-standby`을 사용하는 방법을 보여 줍니다.

AWS CLI

대기 모드에서 인스턴스를 이동하려면

이 예제에서는 지정된 인스턴스를 대기 모드에서 벗어납니다.

```
aws autoscaling exit-standby \
    --instance-ids i-061c63c5eb45f0416 \
    --auto-scaling-group-name my-asg
```

출력:

```
{
    "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\"}"
        }
    ]
}
```

자세한 내용은 Amazon Auto Scaling 사용 설명서의 Auto Scaling 그룹에서 인스턴스를 일시적으로 제거하는 단원을 참조하세요. EC2 Auto Scaling

- 자세한 API 내용은 명령 참조ExitStandby의 섹션을 참조하세요. AWS CLI

### put-lifecycle-hook

다음 코드 예시에서는 put-lifecycle-hook을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 수명 주기 후크 생성

이 예제에서는 4,800초의 제한 시간으로 새로 시작된 인스턴스에서 호출할 수명 주기 후크를 생성합니다. 이는 사용자 데이터 스크립트가 완료될 때까지 인스턴스를 대기 상태로 유지하거나 를 사용하여 AWS Lambda 함수를 호출하는 데 유용합니다 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
```

이 명령은 출력을 생성하지 않습니다. 동일한 이름의 수명 주기 후크가 이미 있는 경우 새 수명 주기 후크로 덮어씁니다.

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 Amazon Auto Scaling 수명 주기 후크를 참조하세요. EC2 Auto Scaling

예제 2: 인스턴스 상태 전환을 알리기 위해 Amazon SNS 이메일 메시지를 보내려면

이 예제에서는 인스턴스 시작 시 알림을 수신하는 데 사용할 Amazon SNS 주제 및 IAM 역할과 함께 수명 주기 후크를 생성합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 Amazon Auto Scaling 수명 주기 후크를 참조하세요. EC2 Auto Scaling

예제 3: Amazon SQS 대기열에 메시지를 게시하는 방법

이 예제에서는 메타데이터가 포함된 메시지를 지정된 Amazon SQS 대기열에 게시하는 수명 주기 후크를 생성합니다.

```
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"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 Amazon Auto Scaling 수명 주기 후크를 참조하세요. EC2 Auto Scaling

- 자세한 API 내용은 명령 참조PutLifecycleHook의 섹션을 참조하세요. AWS CLI

### put-notification-configuration

다음 코드 예시에서는 put-notification-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

알림을 추가하려면

이 예제에서는 지정된 Auto Scaling 그룹에 지정된 알림을 추가합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Auto Scaling 사용 설명서의 Auto Scaling 그룹이 확장될 때 Amazon SNS 알림 받기를 참조하세요.  EC2 Auto Scaling

- 자세한 API 내용은 명령 참조PutNotificationConfiguration의 섹션을 참조하세요. AWS CLI

### put-scaling-policy

다음 코드 예시에서는 put-scaling-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

Auto Scaling 그룹에 대상 추적 조정 정책을 추가하려면

다음 put-scaling-policy 예제에서는 지정된 Auto Scaling 그룹에 대상 추적 조정 정책을 적용합니다. 출력에는 사용자를 대신하여 생성된 두 CloudWatch 경보의 ARNs 및 이름이 포함됩니다. 이름이 같은 조정 정책이 이미 있는 경우 새 조정 정책으로 덮어씁니다.

```
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
```

config.json의 콘텐츠:

```
{
     "TargetValue": 1000.0,
     "PredefinedMetricSpecification": {
          "PredefinedMetricType": "ALBRequestCountPerTarget",
          "ResourceLabel": "app/my-alb/778d41231b141a0f/targetgroup/my-alb-target-
group/943f017f100becff"
     }
}
```

출력:

```
{
     "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"
         }
     ]
 }
```

자세한 예는 Amazon EC2 Auto Scaling 사용 설명서의 AWS 명령줄 인터페이스(AWS CLI)에 대한 조정 정책 예제를 참조하세요.

- 자세한 API 내용은 명령 참조PutScalingPolicy의 섹션을 참조하세요. AWS CLI

### put-scheduled-update-group-action

다음 코드 예시에서는 put-scheduled-update-group-action을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: Auto Scaling 그룹에 예약된 작업을 추가하려면

이 예제에서는 지정된 Auto Scaling 그룹에 지정된 예약된 작업을 추가합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다. 동일한 이름의 예약된 작업이 이미 있는 경우 새 예약된 작업으로 덮어씁니다.

자세한 예는 Amazon EC2 Auto Scaling 사용 설명서의 예약된 조정을 참조하세요.

예제 2: 반복 일정을 지정하려면

이 예제에서는 매년 1월, 6월, 12월 1일에 00:30시에 실행되도록 예약된 작업을 생성하여 반복 일정에 따라 확장합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다. 동일한 이름의 예약된 작업이 이미 있는 경우 새 예약된 작업으로 덮어씁니다.

자세한 예는 Amazon EC2 Auto Scaling 사용 설명서의 예약된 조정을 참조하세요.

- 자세한 API 내용은 명령 참조PutScheduledUpdateGroupAction의 섹션을 참조하세요. AWS CLI

### put-warm-pool

다음 코드 예시에서는 put-warm-pool을 사용하는 방법을 보여 줍니다.

AWS CLI

웜 풀을 생성하려면

다음 예제에서는 지정된 Auto Scaling 그룹에 대한 웜 풀을 생성합니다.

```
aws autoscaling put-warm-pool \
    --auto-scaling-group-name my-asg \
    --min-size 2
```

이 명령은 출력을 생성하지 않습니다. 웜 풀이 이미 있는 경우 업데이트됩니다.

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 Amazon Auto Scaling용 웜 풀을 참조하세요.  EC2 Auto Scaling

- 자세한 API 내용은 명령 참조PutWarmPool의 섹션을 참조하세요. AWS CLI

### `record-lifecycle-action-heartbeat`

다음 코드 예시에서는 `record-lifecycle-action-heartbeat`을 사용하는 방법을 보여 줍니다.

AWS CLI

수명 주기 작업 하트비트를 기록하려면

이 예제에서는 인스턴스를 보류 상태로 유지하기 위해 수명 주기 작업 하트비트를 기록합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 Amazon Auto Scaling 수명 주기 후크를 참조하세요.  EC2 Auto Scaling

- 자세한 API 내용은 명령 참조RecordLifecycleActionHeartbeat의 섹션을 참조하세요. AWS CLI

### `resume-processes`

다음 코드 예시에서는 `resume-processes`을 사용하는 방법을 보여 줍니다.

AWS CLI

일시 중지된 프로세스를 재개하려면

이 예제에서는 지정된 Auto Scaling 그룹에 대해 지정된 일시 중지된 조정 프로세스를 재개합니다.

```
aws autoscaling resume-processes \
    --auto-scaling-group-name my-asg \
    --scaling-processes AlarmNotification
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 조정 프로세스 일시 중지 및 재개를 참조하세요.

- 자세한 API 내용은 명령 참조ResumeProcesses의 섹션을 참조하세요. AWS CLI

### `rollback-instance-refresh`

다음 코드 예시에서는 `rollback-instance-refresh`을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스 새로 고침을 롤백하려면

다음 `rollback-instance-refresh` 예제에서는 지정된 Auto Scaling 그룹에 대해 진행 중인 인스턴스 새로 고침을 롤백합니다.

```
aws autoscaling rollback-instance-refresh \
    --auto-scaling-group-name my-asg
```

출력:

```
{
    "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"
}
```

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 롤백으로 변경 사항 실행 취소를 참조하세요.

- 자세한 API 내용은 명령 참조RollbackInstanceRefresh의 섹션을 참조하세요. AWS CLI

### `set-desired-capacity`

다음 코드 예시에서는 `set-desired-capacity`을 사용하는 방법을 보여 줍니다.

AWS CLI

Auto Scaling 그룹에 원하는 용량을 설정하는 방법

이 예시에서는 지정된 Auto Scaling 그룹에 원하는 용량을 설정합니다.

```
aws autoscaling set-desired-capacity \
    --auto-scaling-group-name my-asg \
    --desired-capacity 2 \
    --honor-cooldown
```

이 명령은 성공하면 프롬프트로 돌아갑니다.

- 자세한 API 내용은 명령 참조SetDesiredCapacity의 섹션을 참조하세요. AWS CLI

### set-instance-health

다음 코드 예시에서는 set-instance-health을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스의 상태를 설정하려면

이 예제에서는 지정된 인스턴스의 상태를 로 설정합니다Unhealthy.

```
aws autoscaling set-instance-health \
    --instance-id i-061c63c5eb45f0416 \
    --health-status Unhealthy
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조SetInstanceHealth의 섹션을 참조하세요. AWS CLI

### set-instance-protection

다음 코드 예시에서는 set-instance-protection을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 인스턴스에 대한 인스턴스 보호 설정을 활성화하려면

이 예제에서는 지정된 인스턴스에 대한 인스턴스 보호를 활성화합니다.

```
aws autoscaling set-instance-protection \
    --instance-ids i-061c63c5eb45f0416 \
    --auto-scaling-group-name my-asg --protected-from-scale-in
```

이 명령은 출력을 생성하지 않습니다.

예제 2: 인스턴스에 대한 인스턴스 보호 설정을 비활성화하려면

이 예제에서는 지정된 인스턴스에 대한 인스턴스 보호를 비활성화합니다.

```
aws autoscaling set-instance-protection \
```

```
    --instance-ids i-061c63c5eb45f0416 \
    --auto-scaling-group-name my-asg \
    --no-protected-from-scale-in
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조SetInstanceProtection의 섹션을 참조하세요. AWS CLI

### start-instance-refresh

다음 코드 예시에서는 start-instance-refresh을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 명령줄 파라미터를 사용하여 인스턴스 새로 고침을 시작하려면

다음 start-instance-refresh 예제에서는 명령줄 인수를 사용하여 인스턴스 새로 고침을 시작합니다. 선택적 preferences 파라미터는 60 초InstanceWarmup의 와 50 퍼센트MinHealthyPercentage의 를 지정합니다.

```
aws autoscaling start-instance-refresh \
    --auto-scaling-group-name my-asg \
    --preferences '{"InstanceWarmup": 60, "MinHealthyPercentage": 50}'
```

출력:

```
{
    "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"
}
```

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 인스턴스 새로 고침 시작을 참조하세요.

예제 2: JSON 파일을 사용하여 인스턴스 새로 고침을 시작하려면

다음 start-instance-refresh 예제에서는 JSON 파일을 사용하여 인스턴스 새로 고침을 시작합니다. 다음 예제와 같이 Auto Scaling 그룹을 지정하고 원하는 구성 및 기본 설정을 JSON 파일에 정의할 수 있습니다.

```
aws autoscaling start-instance-refresh \
    --cli-input-json file://config.json
```

config.json의 콘텐츠:

```
{
    "AutoScalingGroupName": "my-asg",
    "DesiredConfiguration": {
        "LaunchTemplate": {
            "LaunchTemplateId": "lt-068f72b729example",
            "Version": "$Default"
        }
    },
    "Preferences": {
        "InstanceWarmup": 60,
        "MinHealthyPercentage": 50,
        "AutoRollback": true,
        "ScaleInProtectedInstances": Ignore,
        "StandbyInstances": Terminate
    }
}
```

출력:

```
{
    "InstanceRefreshId": "08b91cf7-8fa6-48af-b6a6-d227f40f1b9b"
}
```

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 인스턴스 새로 고침 시작을 참조하세요.

- 자세한 API 내용은 명령 참조StartInstanceRefresh의 섹션을 참조하세요. AWS CLI

### `suspend-processes`

다음 코드 예시에서는 `suspend-processes`을 사용하는 방법을 보여 줍니다.

AWS CLI

Auto Scaling 프로세스를 일시 중지하려면

이 예제에서는 지정된 Auto Scaling 그룹에 대해 지정된 조정 프로세스를 일시 중지합니다.

```
aws autoscaling suspend-processes \
    --auto-scaling-group-name my-asg \
    --scaling-processes AlarmNotification
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 조정 프로세스 일시 중지 및 재개를 참조하세요.

- 자세한 API 내용은 명령 참조SuspendProcesses의 섹션을 참조하세요. AWS CLI

### terminate-instance-in-auto-scaling-group

다음 코드 예시에서는 terminate-instance-in-auto-scaling-group을 사용하는 방법을 보여줍니다.

AWS CLI

Auto Scaling 그룹에서 인스턴스를 종료하는 방법

이 예시에서는 그룹 크기를 업데이트하지 않고 지정된 Auto Scaling 그룹에서 지정된 인스턴스를 종료합니다. Amazon EC2 Auto Scaling은 지정된 인스턴스가 종료된 후 대체 인스턴스를 시작합니다.

```
aws autoscaling terminate-instance-in-auto-scaling-group \
    --instance-id i-061c63c5eb45f0416 \
    --no-should-decrement-desired-capacity
```

출력:

```
{
    "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\"}"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조TerminateInstanceInAutoScalingGroup의 섹션을 참조하세요. AWS CLI

### update-auto-scaling-group

다음 코드 예시에서는 update-auto-scaling-group을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: Auto Scaling 그룹의 크기 한도를 업데이트하는 방법

이 예시에서는 지정된 Auto Scaling 그룹을 최소 크기가 2, 최대 크기가 10으로 업데이트합니다.

```
aws autoscaling update-auto-scaling-group \
    --auto-scaling-group-name my-asg \
    --min-size 2 \
    --max-size 10
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Auto Scaling 사용 설명서의 Auto Scaling 그룹의 용량 제한 설정을 참조하세요. EC2 Auto Scaling

예 2: Elastic Load Balancing 상태 확인을 추가하고 사용할 가용 영역 및 서브넷을 지정하는 방법

이 예시에서는 Elastic Load Balancing 상태 확인을 추가하도록 지정된 Auto Scaling 그룹을 업데이트합니다. 또한 이 명령은 의 값을 IDs 여러 가용 영역의 서브넷 목록--vpc-zone-identifier으로 업데이트합니다.

```
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"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 Elastic Load Balancing 및 Amazon Auto Scaling을 참조하세요. EC2 Auto Scaling

예 3: 배치 그룹 및 종료 정책을 업데이트하는 방법

이 예시에서는 사용할 배치 그룹 및 종료 정책을 업데이트합니다.

```
aws autoscaling update-auto-scaling-group \
```

```
    --auto-scaling-group-name my-asg \
    --placement-group my-placement-group \
    --termination-policies "OldestInstance"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Auto Scaling 사용 설명서의 Auto Scaling 그룹을 참조하세요. EC2 Auto Scaling

예 4: 시작 템플릿의 최신 버전을 사용하는 방법

이 예시에서는 지정된 시작 템플릿의 최신 버전을 사용하도록 지정된 Auto Scaling 그룹을 업데이트합니다.

```
aws autoscaling update-auto-scaling-group \
    --auto-scaling-group-name my-asg \
    --launch-template LaunchTemplateId=lt-1234567890abcde12,Version='$Latest'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 시작 템플릿을 참조하세요.

예 5: 시작 템플릿의 특정 버전을 사용하는 방법

이 예시에서는 시작 템플릿의 최신 또는 기본 버전 대신 특정 버전을 사용하도록 지정된 Auto Scaling 그룹을 업데이트합니다.

```
aws autoscaling update-auto-scaling-group \
    --auto-scaling-group-name my-asg \
    --launch-template LaunchTemplateName=my-template-for-auto-scaling,Version='2'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon EC2 Auto Scaling 사용 설명서의 시작 템플릿을 참조하세요.

예 6: 혼합 인스턴스 정책을 정의하고 용량 재분배를 활성화하는 방법

이 예시에서는 혼합 인스턴스 정책을 사용하도록 지정된 Auto Scaling 그룹을 업데이트하고 용량 재분배를 활성화합니다. 이 구조를 통해 스팟 및 온디맨드 용량을 사용하는 그룹을 지정하고 아키텍처마다 다른 시작 템플릿을 사용할 수 있습니다.

```
aws autoscaling update-auto-scaling-group \
    --cli-input-json file://~/config.json
```

config.json의 콘텐츠:

```
{
    "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"
        }
    }
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Auto Scaling 사용 설명서의 여러 인스턴스 유형 및 구매 옵션이 있는 Auto Scaling 그룹을 참조하세요.  EC2 Auto Scaling

- 자세한 API 내용은 명령 참조UpdateAutoScalingGroup의 섹션을 참조하세요. AWS CLI

# 를 사용한 Auto Scaling Plans 예제 AWS CLI

다음 코드 예제에서는 Auto Scaling Plans와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `create-scaling-plan`

다음 코드 예시에서는 `create-scaling-plan`을 사용하는 방법을 보여 줍니다.

AWS CLI

조정 계획을 생성하려면

다음 `create-scaling-plan` 예제에서는 이미 생성된 JSON 파일(config.json 이름)을 `my-scaling-plan` 사용하여 라는 조정 계획을 생성합니다. 조정 계획의 구조에는 라는 Auto Scaling 그룹에 대한 조정 지침이 포함되어 있습니다`my-asg`. 이 계획은 `TagFilters` 속성을 애플리케이션 소스로 지정하고 예측 조정 및 동적 조정을 활성화합니다.

```
aws autoscaling-plans create-scaling-plan \
    --scaling-plan-name my-scaling-plan \
    --cli-input-json file://~/config.json
```

`config.json` 파일의 콘텐츠:

```
{
    "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
                }
            ]
        }
    ]
}
```

출력:

```
{
"ScalingPlanVersion": 1
}
```

자세한 내용은 AWS Auto Scaling 사용 설명서 를 참조하세요.

- 자세한 API 내용은 명령 참조CreateScalingPlan의 섹션을 참조하세요. AWS CLI

### delete-scaling-plan

다음 코드 예시에서는 delete-scaling-plan을 사용하는 방법을 보여 줍니다.

AWS CLI

조정 계획을 삭제하려면

다음 delete-scaling-plan 예제에서는 지정된 조정 계획을 삭제합니다.

```
aws autoscaling-plans delete-scaling-plan \
    --scaling-plan-name my-scaling-plan \
    --scaling-plan-version 1
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Auto Scaling 사용 설명서 를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteScalingPlan의 섹션을 참조하세요. AWS CLI

### describe-scaling-plan-resources

다음 코드 예시에서는 describe-scaling-plan-resources을 사용하는 방법을 보여 줍니다.

AWS CLI

규모 조정 계획을 위한 확장 가능 리소스를 설명하려면

다음 describe-scaling-plan-resources 예제에서는 지정된 조정 계획과 연결된 단일 확장 가능 리소스(Auto Scaling 그룹)에 대한 세부 정보를 표시합니다.

```
aws autoscaling-plans describe-scaling-plan-resources \
    --scaling-plan-name my-scaling-plan \
    --scaling-plan-version 1
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS Auto Scaling 사용 설명서의 Auto Scaling이란 무엇입니까?를 참조하세요.

- AWS Auto Scaling
- 자세한 API 내용은 명령 참조DescribeScalingPlanResources의 섹션을 참조하세요. AWS CLI

### `describe-scaling-plans`

다음 코드 예시에서는 `describe-scaling-plans`을 사용하는 방법을 보여 줍니다.

AWS CLI

조정 계획을 설명하려면

다음 `describe-scaling-plans` 예제에서는 지정된 조정 계획의 세부 정보를 표시합니다.

```
aws autoscaling-plans describe-scaling-plans \
    --scaling-plan-names scaling-plan-with-asg-and-ddb
```

출력:

```
{
    "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"
```

```
        }
    ]
}
```

자세한 내용은 AWS Auto Scaling 사용 설명서의 Auto Scaling이란 무엇입니까?를 참조하세요. AWS Auto Scaling

- 자세한 API 내용은 명령 참조DescribeScalingPlans의 섹션을 참조하세요. AWS CLI

### `get-scaling-plan-resource-forecast-data`

다음 코드 예시에서는 `get-scaling-plan-resource-forecast-data`을 사용하는 방법을 보여 줍니다.

AWS CLI

로드 예측 데이터를 검색하려면

이 예제에서는 지정된 조정 계획과 연결된 확장 가능 리소스(Auto Scaling 조정 그룹)에 대한 로드 예측 데이터를 검색합니다.

```
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"
```

출력:

```
{
    "Datapoints": [...]
}
```

자세한 내용은 AWS Auto Scaling 사용 설명서AWS 의 IsAuto Scaling 항목을 참조하세요.

- 자세한 API 내용은 명령 참조GetScalingPlanResourceForecastData의 섹션을 참조하세요. AWS CLI

**update-scaling-plan**

다음 코드 예시에서는 update-scaling-plan을 사용하는 방법을 보여 줍니다.

AWS CLI

조정 계획을 업데이트하려면

다음 update-scaling-plan 예제에서는 지정된 조정 계획에서 Auto Scaling 그룹에 대한 조정 지표를 수정합니다.

```
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}'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Auto Scaling 사용 설명서의 Auto Scaling이란 무엇입니까?를 참조하세요. AWS Auto Scaling

- 자세한 API 내용은 명령 참조UpdateScalingPlan의 섹션을 참조하세요. AWS CLI

## AWS Backup 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS Backup.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### create-backup-plan

다음 코드 예시에서는 create-backup-plan을 사용하는 방법을 보여 줍니다.

AWS CLI

백업 계획을 생성하려면

다음 create-backup-plan 예제에서는 35일 보존을 사용하여 지정된 백업 계획을 생성합니다.

```
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}}]}"
```

출력:

```
{
    "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": "ZjQ2ZTI5YWQtZDg5Yi00MzYzLWJmZTAtMDI1MzhlMDhjYjEz"
}
```

자세한 내용은 백업 개발자 안내서의 백업 계획 생성을 참조하세요. AWS

- 자세한 API 내용은 명령 참조CreateBackupPlan의 섹션을 참조하세요. AWS CLI

### create-backup-vault

다음 코드 예시에서는 create-backup-vault을 사용하는 방법을 보여 줍니다.

AWS CLI

백업 볼트를 생성하려면

다음 create-backup-vault 예제에서는 지정된 이름으로 백업 볼트를 생성합니다.

```
aws backup create-backup-vault
    --backup-vault-name sample-vault
```

이 명령은 출력을 생성하지 않습니다. 출력:

```
{
    "BackupVaultName": "sample-vault",
    "BackupVaultArn": "arn:aws:backup:us-west-2:123456789012:backup-vault:sample-
vault",
    "CreationDate": 1568928338.385
}
```

자세한 내용은 백업 개발자 안내서의 백업 볼트 생성을 참조하세요. AWS

- 자세한 API 내용은 명령 참조CreateBackupVault의 섹션을 참조하세요. AWS CLI

### get-backup-plan-from-template

다음 코드 예시에서는 get-backup-plan-from-template을 사용하는 방법을 보여 줍니다.

AWS CLI

템플릿에서 기존 백업 계획을 가져오려면

다음 get-backup-plan-from-template 예제에서는 35일 보존 기간을 가진 일일 백업을 지정하는 템플릿에서 기존 백업 계획을 가져옵니다.

```
aws backup get-backup-plan-from-template \
    --backup-plan-template-id "87c0c1ef-254d-4180-8fef-2e76a2c38aaa"
```

출력:

```
{
    "BackupPlanDocument": {
        "Rules": [
            {
                "RuleName": "DailyBackups",
                "ScheduleExpression": "cron(0 5 ? * * *)",
                "StartWindowMinutes": 480,
                "Lifecycle": {
                    "DeleteAfterDays": 35
```

```
                }
            }
        ]
    }
}
```

자세한 내용은 백업 개발자 안내서의 백업 계획 생성을 참조하세요. AWS

- 자세한 API 내용은 명령 참조GetBackupPlanFromTemplate의 섹션을 참조하세요. AWS CLI

### get-backup-plan

다음 코드 예시에서는 get-backup-plan을 사용하는 방법을 보여 줍니다.

AWS CLI

백업 계획의 세부 정보를 가져오려면

다음 get-backup-plan 예제에서는 지정된 백업 계획의 세부 정보를 표시합니다.

```
aws backup get-backup-plan \
    --backup-plan-id "fcbf5d8f-bd77-4f3a-9c97-f24fb3d373a5"
```

출력:

```
{
    "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": "NjQ2ZTZkODktMGVhNy00MmQ0LWE4YjktZTkwNTQ3OTkyYTcw",
    "CreationDate": 1568926091.57
}
```

자세한 내용은 백업 개발자 안내서의 백업 계획 생성을 참조하세요. AWS

- 자세한 API 내용은 명령 참조GetBackupPlan의 섹션을 참조하세요. AWS CLI

### `list-backup-jobs`

다음 코드 예시에서는 `list-backup-jobs`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 모든 백업 작업을 나열하려면

다음 `list-backup-jobs` 예제에서는 AWS 계정의 백업 작업에 대한 메타데이터를 반환합니다.

```
aws backup list-backup-jobs
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 백업 개발자 안내서의 백업 생성을 참조하세요. AWS

예제 2: 완료된 백업 작업을 나열하려면

다음 `list-backup-jobs` 예제에서는 AWS 계정에서 완료된 백업 작업에 대한 메타데이터를 반환합니다.

```
aws backup list-backup-jobs \
    --by-state COMPLETED
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 백업 개발자 안내서의 백업 생성을 참조하세요. AWS

- 자세한 API 내용은 명령 참조ListBackupJobs의 섹션을 참조하세요. AWS CLI

## AWS Batch 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS Batch.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `cancel-job`

다음 코드 예시에서는 `cancel-job`을 사용하는 방법을 보여 줍니다.

AWS CLI

작업을 취소하려면

이 예제에서는 지정된 작업 ID로 작업을 취소합니다.

명령:

```
aws batch cancel-job --job-id bcf0b186-a532-4122-842e-2ccab8d54efb --
reason "Cancelling job."
```

- 자세한 API 내용은 명령 참조CancelJob의 섹션을 참조하세요. AWS CLI

### `create-compute-environment`

다음 코드 예시에서는 `create-compute-environment`을 사용하는 방법을 보여 줍니다.

AWS CLI

온디맨드 인스턴스를 사용하여 관리형 컴퓨팅 환경을 생성하려면

이 예제에서는 온디맨드 방식으로 시작되는 특정 C4 인스턴스 유형을 사용하여 관리형 컴퓨팅 환경을 생성합니다. 컴퓨팅 환경을 C4OnDemand라고 합니다.

명령:

```
aws batch create-compute-environment --cli-input-json file://<path_to_json_file>/
C4OnDemand.json
```

JSON 파일 형식:

```
{
  "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"
}
```

출력:

```
{
    "computeEnvironmentName": "C4OnDemand",
    "computeEnvironmentArn": "arn:aws:batch:us-east-1:012345678910:compute-
environment/C4OnDemand"
}
```

스팟 인스턴스를 사용하여 관리형 컴퓨팅 환경을 생성하려면

이 예제에서는 스팟 입찰 가격이 인스턴스 유형에 대한 온디맨드 가격의 20% 이하일 때 시작되는 M4 인스턴스 유형을 사용하여 관리형 컴퓨팅 환경을 생성합니다. 컴퓨팅 환경을 M4Spot 이라고 합니다.

명령:

```
aws batch create-compute-environment --cli-input-json file://<path_to_json_file>/
M4Spot.json
```

JSON 파일 형식:

```
{
  "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"
}
```

출력:

```
{
    "computeEnvironmentName": "M4Spot",
    "computeEnvironmentArn": "arn:aws:batch:us-east-1:012345678910:compute-
environment/M4Spot"
}
```

- 자세한 API 내용은 명령 참조CreateComputeEnvironment의 섹션을 참조하세요. AWS CLI

### `create-job-queue`

다음 코드 예시에서는 `create-job-queue`을 사용하는 방법을 보여 줍니다.

AWS CLI

단일 컴퓨팅 환경으로 우선 순위가 낮은 작업 대기열을 생성하려면

이 예제에서는 M4Spot 컴퓨팅 환경을 LowPriority 사용하는 라는 작업 대기열을 생성합니다.

명령:

```
aws batch create-job-queue --cli-input-json file://<path_to_json_file>/
LowPriority.json
```

JSON 파일 형식:

```
{
  "jobQueueName": "LowPriority",
  "state": "ENABLED",
  "priority": 10,
  "computeEnvironmentOrder": [
    {
      "order": 1,
      "computeEnvironment": "M4Spot"
    }
  ]
}
```

출력:

```
{
    "jobQueueArn": "arn:aws:batch:us-east-1:012345678910:job-queue/LowPriority",
    "jobQueueName": "LowPriority"
}
```

두 컴퓨팅 환경으로 높은 우선 순위 작업 대기열을 생성하려면

이 예제에서는 순서 HighPriority 가 1인 C4OnDemand compute 환경과 순서가 2인 M4Spot 컴퓨팅 환경을 사용하는 라는 작업 대기열을 생성합니다. 스케줄러는 먼저 C4OnDemand compute 환경에 작업을 배치하려고 시도합니다.

명령:

```
aws batch create-job-queue --cli-input-json file://<path_to_json_file>/
HighPriority.json
```

JSON 파일 형식:

```
{
  "jobQueueName": "HighPriority",
  "state": "ENABLED",
  "priority": 1,
  "computeEnvironmentOrder": [
    {
      "order": 1,
      "computeEnvironment": "C4OnDemand"
    },
    {
      "order": 2,
      "computeEnvironment": "M4Spot"
    }
  ]
}
```

출력:

```
{
    "jobQueueArn": "arn:aws:batch:us-east-1:012345678910:job-queue/HighPriority",
    "jobQueueName": "HighPriority"
}
```

- 자세한 API 내용은 명령 참조CreateJobQueue의 섹션을 참조하세요. AWS CLI

### `delete-compute-environment`

다음 코드 예시에서는 `delete-compute-environment`을 사용하는 방법을 보여 줍니다.

AWS CLI

컴퓨팅 환경을 삭제하려면

이 예제에서는 P2OnDemand compute 환경을 삭제합니다.

명령:

```
aws batch delete-compute-environment --compute-environment P2OnDemand
```

- 자세한 API 내용은 명령 참조DeleteComputeEnvironment의 섹션을 참조하세요. AWS CLI

### delete-job-queue

다음 코드 예시에서는 delete-job-queue을 사용하는 방법을 보여 줍니다.

AWS CLI

작업 대기열을 삭제하려면

이 예제에서는 GPGPU 작업 대기열을 삭제합니다.

명령:

```
aws batch delete-job-queue --job-queue GPGPU
```

- 자세한 API 내용은 명령 참조DeleteJobQueue의 섹션을 참조하세요. AWS CLI

### deregister-job-definition

다음 코드 예시에서는 deregister-job-definition을 사용하는 방법을 보여 줍니다.

AWS CLI

작업 정의를 등록 취소하려면

이 예제에서는 sleep10이라는 작업 정의를 등록 취소합니다.

명령:

```
aws batch deregister-job-definition --job-definition sleep10
```

- 자세한 API 내용은 명령 참조DeregisterJobDefinition의 섹션을 참조하세요. AWS CLI

### describe-compute-environments

다음 코드 예시에서는 describe-compute-environments을 사용하는 방법을 보여 줍니다.

AWS CLI

컴퓨팅 환경을 설명하는 방법

이 예제에서는 P2OnDemand compute 환경을 설명합니다.

명령:

```
aws batch describe-compute-environments --compute-environments P2OnDemand
```

출력:

```
{
    "computeEnvironments": [
        {
            "status": "VALID",
            "serviceRole": "arn:aws:iam::012345678910:role/AWSBatchServiceRole",
            "computeEnvironmentArn": "arn:aws:batch:us-east-1:012345678910:compute-
environment/P2OnDemand",
            "computeResources": {
                "subnets": [
                    "subnet-220c0e0a",
                    "subnet-1a95556d",
                    "subnet-978f6dce"
                ],
                "tags": {
                    "Name": "Batch Instance - P2OnDemand"
                },
                "desiredvCpus": 48,
                "minvCpus": 0,
                "instanceTypes": [
                    "p2"
                ],
                "securityGroupIds": [
                    "sg-cf5093b2"
                ],
                "instanceRole": "ecsInstanceRole",
                "maxvCpus": 128,
                "type": "EC2",
                "ec2KeyPair": "id_rsa"
            },
            "statusReason": "ComputeEnvironment Healthy",
            "ecsClusterArn": "arn:aws:ecs:us-east-1:012345678910:cluster/
P2OnDemand_Batch_2c06f29d-d1fe-3a49-879d-42394c86effc",
            "state": "ENABLED",
            "computeEnvironmentName": "P2OnDemand",
            "type": "MANAGED"
```

```
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeComputeEnvironments의 섹션을 참조하세요. AWS CLI

### `describe-job-definitions`

다음 코드 예시에서는 `describe-job-definitions`을 사용하는 방법을 보여 줍니다.

AWS CLI

활성 작업 정의를 설명하려면

이 예제에서는 모든 활성 작업 정의를 설명합니다.

명령:

```
aws batch describe-job-definitions --status ACTIVE
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeJobDefinitions의 섹션을 참조하세요. AWS CLI

### `describe-job-queues`

다음 코드 예시에서는 `describe-job-queues`을 사용하는 방법을 보여 줍니다.

AWS CLI

작업 대기열을 설명하려면

이 예제에서는 HighPriority 작업 대기열을 설명합니다.

명령:

```
aws batch describe-job-queues --job-queues HighPriority
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeJobQueues의 섹션을 참조하세요. AWS CLI

### `describe-jobs`

다음 코드 예시에서는 `describe-jobs`을 사용하는 방법을 보여 줍니다.

AWS CLI

작업을 설명하려면

다음 `describe-jobs` 예제에서는 지정된 작업 ID를 가진 작업에 대해 설명합니다.

```
aws batch describe-jobs \
    --jobs bcf0b186-a532-4122-842e-2ccab8d54efb
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeJobs의 섹션을 참조하세요. AWS CLI

### `list-jobs`

다음 코드 예시에서는 `list-jobs`을 사용하는 방법을 보여 줍니다.

AWS CLI

실행 중인 작업을 나열하려면

이 예제에서는 작업 대기열에서 실행 중인 HighPriority 작업을 나열합니다.

명령:

```
aws batch list-jobs --job-queue HighPriority
```

출력:

```
{
    "jobSummaryList": [
        {
            "jobName": "example",
            "jobId": "e66ff5fd-a1ff-4640-b1a2-0b0a142f49bb"
        }
    ]
}
```

제출된 작업을 나열하려면

이 예제에서는 작업 대기열에서 HighPriority 작업 상태에 있는 SUBMITTED 작업을 나열합니다.

명령:

```
aws batch list-jobs --job-queue HighPriority --job-status SUBMITTED
```

출력:

```
{
    "jobSummaryList": [
        {
            "jobName": "example",
            "jobId": "68f0c163-fbd4-44e6-9fd1-25b14a434786"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListJobs의 섹션을 참조하세요. AWS CLI

### register-job-definition

다음 코드 예시에서는 register-job-definition을 사용하는 방법을 보여 줍니다.

AWS CLI

작업 정의를 등록하려면

이 예제에서는 간단한 컨테이너 작업에 대한 작업 정의를 등록합니다.

명령:

```
aws batch register-job-definition --job-definition-name sleep30 --type container --
container-properties '{ "image": "busybox", "vcpus": 1, "memory": 128, "command":
 [ "sleep", "30"]}'
```

출력:

```
{
    "jobDefinitionArn": "arn:aws:batch:us-east-1:012345678910:job-definition/
sleep30:1",
    "jobDefinitionName": "sleep30",
    "revision": 1
}
```

- 자세한 API 내용은 명령 참조RegisterJobDefinition의 섹션을 참조하세요. AWS CLI

### submit-job

다음 코드 예시에서는 submit-job을 사용하는 방법을 보여 줍니다.

AWS CLI

작업을 제출하려면

이 예제에서는 예제라는 간단한 컨테이너 작업을 HighPriority 작업 대기열에 제출합니다.

명령:

```
aws batch submit-job --job-name example --job-queue HighPriority  --job-
definition sleep60
```

출력:

```
{
    "jobName": "example",
    "jobId": "876da822-4198-45f2-a252-6cea32512ea8"
}
```

- 자세한 API 내용은 명령 참조SubmitJob의 섹션을 참조하세요. AWS CLI

### terminate-job

다음 코드 예시에서는 terminate-job을 사용하는 방법을 보여 줍니다.

AWS CLI

작업을 종료하려면

이 예제에서는 지정된 작업 ID로 작업을 종료합니다.

명령:

```
aws batch terminate-job --job-id 61e743ed-35e4-48da-b2de-5c8333821c84 --
reason "Terminating job."
```

- 자세한 API 내용은 명령 참조TerminateJob의 섹션을 참조하세요. AWS CLI

### update-compute-environment

다음 코드 예시에서는 update-compute-environment을 사용하는 방법을 보여 줍니다.

AWS CLI

컴퓨팅 환경을 업데이트하려면

이 예제에서는 P2OnDemand compute 환경을 비활성화하여 삭제할 수 있습니다.

명령:

```
aws batch update-compute-environment --compute-environment P2OnDemand --
state DISABLED
```

출력:

```
{
    "computeEnvironmentName": "P2OnDemand",
    "computeEnvironmentArn": "arn:aws:batch:us-east-1:012345678910:compute-
environment/P2OnDemand"
}
```

- 자세한 API 내용은 명령 참조UpdateComputeEnvironment의 섹션을 참조하세요. AWS CLI

### update-job-queue

다음 코드 예시에서는 update-job-queue을 사용하는 방법을 보여 줍니다.

AWS CLI

작업 대기열을 업데이트하려면

이 예제에서는 작업 대기열을 비활성화하여 삭제할 수 있도록 합니다.

명령:

```
aws batch update-job-queue --job-queue GPGPU --state DISABLED
```

출력:

```
{
    "jobQueueArn": "arn:aws:batch:us-east-1:012345678910:job-queue/GPGPU",
    "jobQueueName": "GPGPU"
}
```

- 자세한 API 내용은 명령 참조UpdateJobQueue의 섹션을 참조하세요. AWS CLI

## AWS Budgets 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS Budgets.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `create-budget`

다음 코드 예시에서는 `create-budget`을 사용하는 방법을 보여 줍니다.

AWS CLI

비용 및 사용 예산을 생성하려면

다음 `create-budget` 명령은 비용 및 사용 예산을 생성합니다.

```
aws budgets create-budget \
    --account-id 111122223333 \
    --budget file://budget.json \
    --notifications-with-subscribers file://notifications-with-subscribers.json
```

budget.json의 콘텐츠:

```
{
    "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"
}
```

notifications-with-subscribers.json의 콘텐츠:

```
[
    {
        "Notification": {
            "ComparisonOperator": "GREATER_THAN",
            "NotificationType": "ACTUAL",
            "Threshold": 80,
            "ThresholdType": "PERCENTAGE"
```

```
        },
        "Subscribers": [
            {
                "Address": "example@example.com",
                "SubscriptionType": "EMAIL"
            }
        ]
    }
]
```

- 자세한 API 내용은 명령 참조CreateBudget의 섹션을 참조하세요. AWS CLI

### `create-notification`

다음 코드 예시에서는 `create-notification`을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 비용 및 사용 예산에 대한 알림을 생성하려면

이 예제에서는 지정된 비용 및 사용 예산에 대한 알림을 생성합니다.

명령:

```
aws budgets create-notification --account-id 111122223333 --budget-name "Example
 Budget" --
notification NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=80,ThresholdT
 --subscriber SubscriptionType=EMAIL,Address=example@example.com
```

- 자세한 API 내용은 명령 참조CreateNotification의 섹션을 참조하세요. AWS CLI

### `create-subscriber`

다음 코드 예시에서는 `create-subscriber`을 사용하는 방법을 보여 줍니다.

AWS CLI

비용 및 사용 예산과 관련된 알림 구독자를 생성하려면

이 예제에서는 지정된 알림에 대한 구독자를 생성합니다.

명령:

```
aws budgets create-subscriber --account-id 111122223333 --budget-name "Example
 Budget" --
notification NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=80,ThresholdT
 --subscriber SubscriptionType=EMAIL,Address=example@example.com
```

- 자세한 API 내용은 명령 참조CreateSubscriber의 섹션을 참조하세요. AWS CLI

### delete-budget

다음 코드 예시에서는 delete-budget을 사용하는 방법을 보여 줍니다.

AWS CLI

비용 및 사용 예산을 삭제하려면

이 예제에서는 지정된 비용 및 사용 예산을 삭제합니다.

명령:

```
aws budgets delete-budget --account-id 111122223333 --budget-name "Example Budget"
```

- 자세한 API 내용은 명령 참조DeleteBudget의 섹션을 참조하세요. AWS CLI

### delete-notification

다음 코드 예시에서는 delete-notification을 사용하는 방법을 보여 줍니다.

AWS CLI

예산에서 알림을 삭제하려면

이 예제에서는 지정된 예산에서 지정된 알림을 삭제합니다.

명령:

```
aws budgets delete-notification --account-id 111122223333 --budget-name "Example
 Budget" --
notification NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=80,ThresholdT
```

- 자세한 API 내용은 명령 참조DeleteNotification의 섹션을 참조하세요. AWS CLI

### delete-subscriber

다음 코드 예시에서는 delete-subscriber을 사용하는 방법을 보여 줍니다.

AWS CLI

알림에서 구독자를 삭제하려면

이 예제에서는 지정된 알림에서 지정된 구독자를 삭제합니다.

명령:

```
aws budgets delete-subscriber --account-id 111122223333 --budget-name "Example
 Budget" --
notification NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=80,ThresholdT
 --subscriber SubscriptionType=EMAIL,Address=example@example.com
```

- 자세한 API 내용은 명령 참조DeleteSubscriber의 섹션을 참조하세요. AWS CLI

### describe-budget

다음 코드 예시에서는 describe-budget을 사용하는 방법을 보여 줍니다.

AWS CLI

계정과 연결된 예산을 검색하려면

이 예제에서는 지정된 비용 및 사용 예산을 검색합니다.

명령:

```
aws budgets describe-budget --account-id 111122223333 --budget-name "Example Budget"
```

출력:

```
{
   "Budget": {
       "CalculatedSpend": {
           "ForecastedSpend": {
               "Amount": "2641.548000000000229192664846777915954589843 75",
               "Unit": "USD"
```

```
            },
            "ActualSpend": {
                "Amount": "604.456000000000017280399333685636520385742l875",
                "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"
            ]
        }
    }
}
```

- 자세한 API 내용은 명령 참조DescribeBudget의 섹션을 참조하세요. AWS CLI

### `describe-budgets`

다음 코드 예시에서는 `describe-budgets`을 사용하는 방법을 보여 줍니다.

AWS CLI

계정과 연결된 예산을 검색하려면

이 예제에서는 계정의 비용 및 사용 예산을 검색합니다.

명령:

```
aws budgets describe-budgets --account-id 111122223333 --max-results 20
```

출력:

```
{
   "Budgets": [
       {
           "CalculatedSpend": {
               "ForecastedSpend": {
                   "Amount": "2641.548000000000229192664846777915954589843 75",
                   "Unit": "USD"
               },
               "ActualSpend": {
                   "Amount": "604.456000000000017280399333685636520385742 1875",
                   "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"
                ]
            }
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeBudgets의 섹션을 참조하세요. AWS CLI

### describe-notifications-for-budget

다음 코드 예시에서는 describe-notifications-for-budget을 사용하는 방법을 보여 줍니다.

AWS CLI

예산에 대한 알림을 검색하려면

이 예제에서는 비용 및 사용 예산에 대한 알림을 검색합니다.

명령:

```
aws budgets describe-notifications-for-budget --account-id 111122223333 --budget-
name "Example Budget" --max-results 5
```

출력:

```
{
    "Notifications": [
        {
            "Threshold": 80.0,
            "ComparisonOperator": "GREATER_THAN",
            "NotificationType": "ACTUAL"
        }
```

```
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeNotificationsForBudget의 섹션을 참조하세요. AWS CLI

### `describe-subscribers-for-notification`

다음 코드 예시에서는 `describe-subscribers-for-notification`을 사용하는 방법을 보여 줍니다.

AWS CLI

예산 알림을 위해 구독자를 검색하려면

이 예제에서는 비용 및 사용량 예산 알림 구독자를 검색합니다.

명령:

```
aws budgets describe-subscribers-for-notification --
account-id 111122223333 --budget-name "Example Budget" --
notification NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=80,Threshold1
 --max-results 5
```

출력:

```
{
   "Subscribers": [
       {
           "SubscriptionType": "EMAIL",
           "Address": "example2@example.com"
       },
       {
           "SubscriptionType": "EMAIL",
           "Address": "example@example.com"
       }
   ]
}
```

- 자세한 API 내용은 명령 참조DescribeSubscribersForNotification의 섹션을 참조하세요. AWS CLI

### update-budget

다음 코드 예시에서는 update-budget을 사용하는 방법을 보여 줍니다.

AWS CLI

비용 및 사용 예산에 대한 예산을 교체하려면

이 예제에서는 비용 및 사용 예산을 새 예산으로 대체합니다.

명령:

```
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"
}
```

- 자세한 API 내용은 명령 참조UpdateBudget의 섹션을 참조하세요. AWS CLI

### update-notification

다음 코드 예시에서는 update-notification을 사용하는 방법을 보여 줍니다.

AWS CLI

비용 및 사용 예산에 대한 알림을 교체하려면

이 예제에서는 비용 및 사용 예산에 대한 80% 알림을 90% 알림으로 대체합니다.

명령:

```
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
```

- 자세한 API 내용은 명령 참조UpdateNotification의 섹션을 참조하세요. AWS CLI

### update-subscriber

다음 코드 예시에서는 update-subscriber을 사용하는 방법을 보여 줍니다.

AWS CLI

비용 및 사용 예산에 대한 구독자를 교체하려면

이 예제는 비용 및 사용 예산에 대한 구독자를 대체합니다.

명령:

```
aws budgets update-subscriber --account-id 111122223333 --budget-name "Example
 Budget" --
notification NotificationType=ACTUAL,ComparisonOperator=GREATER_THAN,Threshold=80,ThresholdT
```

```
 --old-subscriber SubscriptionType=EMAIL,Address=example@example.com --new-
subscriber SubscriptionType=EMAIL,Address=example2@example.com
```

- 자세한 API 내용은 명령 참조UpdateSubscriber의 섹션을 참조하세요. AWS CLI

## 를 사용한 Amazon Chime 예제 AWS CLI

다음 코드 예제에서는 Amazon Chime과 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `associate-phone-number-with-user`

다음 코드 예시에서는 `associate-phone-number-with-user`을 사용하는 방법을 보여 줍니다.

AWS CLI

전화번호를 사용자와 연결하려면

다음 `associate-phone-number-with-user` 예제에서는 지정된 전화번호를 사용자와 연결합니다.

```
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"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Chime 관리 안내서의 사용자 전화번호 관리를 참조하세요.

- 자세한 API 내용은 명령 참조AssociatePhoneNumberWithUser의 섹션을 참조하세요. AWS CLI

### associate-phone-numbers-with-voice-connector-group

다음 코드 예시에서는 associate-phone-numbers-with-voice-connector-group을 사용하는 방법을 보여 줍니다.

AWS CLI

전화번호를 Amazon Chime Voice Connector 그룹과 연결하려면

다음 associate-phone-numbers-with-voice-connector-group 예제에서는 지정된 전화번호를 Amazon Chime Voice Connector 그룹과 연결합니다.

```
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
```

출력:

```
{
    "PhoneNumberErrors": []
}
```

자세한 내용은 Amazon Chime 관리 안내서의 Amazon Chime Voice Connector 그룹 작업을 참조하세요.

- 자세한 API 내용은 명령 참조AssociatePhoneNumbersWithVoiceConnectorGroup의 섹션을 참조하세요. AWS CLI

### associate-phone-numbers-with-voice-connector

다음 코드 예시에서는 associate-phone-numbers-with-voice-connector을 사용하는 방법을 보여 줍니다.

AWS CLI

전화번호를 Amazon Chime Voice Connector와 연결하려면

다음 associate-phone-numbers-with-voice-connector 예제에서는 지정된 전화번호를 Amazon Chime Voice Connector와 연결합니다.

```
aws chime associate-phone-numbers-with-voice-connector \
    --voice-connector-id abcdef1ghij2klmno3pqr4 \
    --e164-phone-numbers "+12065550100" "+12065550101"
    --force-associate
```

출력:

```
{
    "PhoneNumberErrors": []
}
```

자세한 내용은 Amazon Chime 관리 안내서의 Amazon Chime Voice Connector 작업을 참조하세요.

- 자세한 API 내용은 명령 참조AssociatePhoneNumbersWithVoiceConnector의 섹션을 참조하세요. AWS CLI

### associate-signin-delegate-groups-with-account

다음 코드 예시에서는 associate-signin-delegate-groups-with-account을 사용하는 방법을 보여 줍니다.

AWS CLI

로그인 위임 그룹을 연결하려면

다음 associate-signin-delegate-groups-with-account 예제에서는 지정된 로그인 위임 그룹을 지정된 Amazon Chime 계정과 연결합니다.

```
aws chime associate-signin-delegate-groups-with-account \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --signin-delegate-groups GroupName=my_users
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Chime 관리 안내서의 사용자 액세스 및 권한 관리를 참조하세요.

- 자세한 API 내용은 명령 참조AssociateSigninDelegateGroupsWithAccount의 섹션을 참조하세요. AWS CLI

### batch-create-room-membership

다음 코드 예시에서는 batch-create-room-membership을 사용하는 방법을 보여 줍니다.

AWS CLI

여러 룸 멤버십을 생성하려면

다음 batch-create-room-membership 예제에서는 채팅룸에 여러 사용자를 채팅룸 멤버로 추가합니다. 또한 사용자에게 관리자 및 멤버 역할을 할당합니다.

```
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"
```

출력:

```
{
    "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": []
}
```

자세한 내용은 Amazon Chime 사용 설명서의 채팅룸 생성을 참조하세요.

- 자세한 API 내용은 명령 참조BatchCreateRoomMembership의 섹션을 참조하세요. AWS CLI

### batch-delete-phone-number

다음 코드 예시에서는 batch-delete-phone-number을 사용하는 방법을 보여 줍니다.

AWS CLI

여러 전화번호를 삭제하려면

다음 batch-delete-phone-number 예제에서는 지정된 전화번호를 모두 삭제합니다.

```
aws chime batch-delete-phone-number \
    --phone-number-ids "%2B12065550100" "%2B12065550101"
```

이 명령은 출력을 생성하지 않습니다. 출력:

```
{
    "PhoneNumberErrors": []
}
```

자세한 내용은 Amazon Chime 관리 안내서의 전화번호 작업을 참조하세요.

- 자세한 API 내용은 명령 참조BatchDeletePhoneNumber의 섹션을 참조하세요. AWS CLI

### batch-suspend-user

다음 코드 예시에서는 batch-suspend-user을 사용하는 방법을 보여 줍니다.

AWS CLI

여러 사용자를 일시 중지하려면

다음 batch-suspend-user 예제에서는 지정된 Amazon Chime 계정에서 나열된 사용자를 일시 중지합니다.

```
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"
```

출력:

```
{
```

```
    "UserErrors": []
}
```

- 자세한 API 내용은 명령 참조BatchSuspendUser의 섹션을 참조하세요. AWS CLI

### batch-unsuspend-user

다음 코드 예시에서는 batch-unsuspend-user을 사용하는 방법을 보여 줍니다.

AWS CLI

여러 사용자의 일시 중지를 취소하려면

다음 batch-unsuspend-user 예제에서는 지정된 Amazon Chime 계정의 나열된 사용자에 대한 이전 일시 중지를 제거합니다.

```
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"
```

출력:

```
{
    "UserErrors": []
}
```

- 자세한 API 내용은 명령 참조BatchUnsuspendUser의 섹션을 참조하세요. AWS CLI

### batch-update-phone-number

다음 코드 예시에서는 batch-update-phone-number을 사용하는 방법을 보여 줍니다.

AWS CLI

여러 전화번호 제품 유형을 동시에 업데이트하려면

다음 batch-update-phone-number 예제에서는 지정된 모든 전화번호의 제품 유형을 업데이트합니다.

```
aws chime batch-update-phone-number \
```

```
    --update-phone-number-request-items PhoneNumberId=
%2B12065550100,ProductType=BusinessCalling PhoneNumberId=
%2B12065550101,ProductType=BusinessCalling
```

출력:

```
{
    "PhoneNumberErrors": []
}
```

여러 전화번호 통화 이름을 동시에 업데이트하려면

다음 batch-update-phone-number 예제에서는 지정된 모든 전화번호의 호출 이름을 업데이트합니다.

```
aws chime batch-update-phone-number \
    --update-phone-number-request-items PhoneNumberId=
%2B14013143874,CallingName=phonenumber1 PhoneNumberId=
%2B14013144061,CallingName=phonenumber2
```

출력:

```
{
    "PhoneNumberErrors": []
}
```

자세한 내용은 Amazon Chime 관리 안내서의 전화번호 작업을 참조하세요.

- 자세한 API 내용은 명령 참조BatchUpdatePhoneNumber의 섹션을 참조하세요. AWS CLI

### batch-update-user

다음 코드 예시에서는 batch-update-user을 사용하는 방법을 보여 줍니다.

AWS CLI

단일 명령으로 여러 사용자를 업데이트하려면

다음 batch-update-user 예제에서는 지정된 Amazon Chime 계정의 나열된 각 사용자에 LicenseType 대해 를 업데이트합니다.

```
aws chime batch-update-user \
```

```
    --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
    --update-user-request-items "UserId=a1b2c3d4-5678-90ab-
cdef-22222EXAMPLE,LicenseType=Basic" "UserId=a1b2c3d4-5678-90ab-
cdef-33333EXAMPLE,LicenseType=Basic"
```

출력:

```
{
    "UserErrors": []
}
```

- 자세한 API 내용은 명령 참조BatchUpdateUser의 섹션을 참조하세요. AWS CLI

### `create-account`

다음 코드 예시에서는 `create-account`을 사용하는 방법을 보여 줍니다.

AWS CLI

계정을 생성하려면

다음 `create-account` 예제에서는 관리자 계정에 Amazon Chime AWS 계정을 생성합니다.

```
aws chime create-account \
    --name MyChimeAccount
```

출력:

```
{
    "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"
            },
        ]
    }
}
```

자세한 내용은 Amazon Chime 관리 안내서의 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조CreateAccount의 섹션을 참조하세요. AWS CLI

### `create-bot`

다음 코드 예시에서는 `create-bot`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon Chime 봇을 생성하려면

다음 `create-bot` 예제에서는 지정된 Amazon Chime Enterprise 계정에 대한 봇을 생성합니다.

```
aws chime create-bot \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --display-name "myBot" \
    --domain "example.com"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Chime 개발자 안내서의 Amazon Chime과 채팅 봇 통합을 참조하세요.

- 자세한 API 내용은 명령 참조CreateBot의 섹션을 참조하세요. AWS CLI

### `create-phone-number-order`

다음 코드 예시에서는 `create-phone-number-order`을 사용하는 방법을 보여 줍니다.

AWS CLI

전화번호 순서를 생성하려면

다음 `create-phone-number-order` 예제에서는 지정된 전화번호에 대한 전화번호 순서를 생성합니다.

```
aws chime create-phone-number-order \
    --product-type VoiceConnector \
    --e164-phone-numbers "+12065550100" "+12065550101" "+12065550102"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Chime 관리 안내서의 전화번호 작업을 참조하세요.

- 자세한 API 내용은 명령 참조CreatePhoneNumberOrder의 섹션을 참조하세요. AWS CLI

### `create-proxy-session`

다음 코드 예시에서는 `create-proxy-session`을 사용하는 방법을 보여 줍니다.

AWS CLI

프록시 세션을 생성하려면

다음 `create-proxy-session` 예제에서는 음성 및 SMS 기능을 사용하여 프록시 세션을 생성합니다.

```
aws chime create-proxy-session \
    --voice-connector-id abcdef1ghij2klmno3pqr4 \
    --participant-phone-numbers "+14015550101" "+12065550100" \
    --capabilities "Voice" "SMS"
```

출력:

```
{
    "ProxySession": {
        "VoiceConnectorId": "abcdef1ghij2klmno3pqr4",
        "ProxySessionId": "123a4bc5-67d8-901e-2f3g-h4ghjk56789l",
        "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"
            }
```

```
        ]
    }
}
```

자세한 내용은 Amazon Chime 개발자 안내서의 프록시 전화 세션을 참조하세요.

- 자세한 API 내용은 명령 참조CreateProxySession의 섹션을 참조하세요. AWS CLI

### `create-room-membership`

다음 코드 예시에서는 `create-room-membership`을 사용하는 방법을 보여 줍니다.

AWS CLI

룸 멤버십을 생성하려면

다음 `create-room-membership` 예제에서는 지정된 사용자를 채팅방에 채팅방 멤버로 추가합니다.

```
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
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Chime 사용 설명서의 채팅룸 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateRoomMembership의 섹션을 참조하세요. AWS CLI

### create-room

다음 코드 예시에서는 create-room을 사용하는 방법을 보여 줍니다.

AWS CLI

채팅방을 생성하려면

다음 create-room 예제에서는 지정된 Amazon Chime 계정에 대한 채팅룸을 생성합니다.

```
aws chime create-room \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --name chatRoom
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Chime 사용 설명서의 채팅룸 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateRoom의 섹션을 참조하세요. AWS CLI

### create-user

다음 코드 예시에서는 create-user을 사용하는 방법을 보여 줍니다.

AWS CLI

공유 디바이스에 대한 사용자 프로필을 생성하려면

다음 create-user 예제에서는 지정된 이메일 주소에 대한 공유 디바이스 프로파일을 생성합니다.

```
aws chime create-user \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --email roomdevice@example.com \
    --user-type SharedDevice
```

출력:

```
{
    "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
        }
    }
}
```

자세한 내용은 Amazon Chime 관리 안내서의 설정 준비를 참조하세요.

- 자세한 API 내용은 명령 참조CreateUser의 섹션을 참조하세요. AWS CLI

### create-voice-connector-group

다음 코드 예시에서는 create-voice-connector-group을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon Chime Voice Connector 그룹을 생성하려면

다음 create-voice-connector-group 예제에서는 지정된 Amazon Chime Voice Connector를 포함하는 Amazon Chime Voice Connector 그룹을 생성합니다.

```
aws chime create-voice-connector-group \
```

```
    --name myGroup \
    --voice-connector-items VoiceConnectorId=abcdef1ghij2klmno3pqr4,Priority=2
```

출력:

```
{
    "VoiceConnectorGroup": {
        "VoiceConnectorGroupId": "123a456b-c7d8-90e1-fg23-4h567jkl8901",
        "Name": "myGroup",
        "VoiceConnectorItems": [],
        "CreatedTimestamp": "2019-09-18T16:38:34.734Z",
        "UpdatedTimestamp": "2019-09-18T16:38:34.734Z"
    }
}
```

자세한 내용은 Amazon Chime 관리 안내서의 Amazon Chime Voice Connector 그룹 작업을 참조하세요.

- 자세한 API 내용은 명령 참조CreateVoiceConnectorGroup의 섹션을 참조하세요. AWS CLI

### create-voice-connector

다음 코드 예시에서는 create-voice-connector을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon Chime Voice Connector를 생성하려면

다음 create-voice-connector 예제에서는 암호화가 활성화된 상태로 지정된 AWS 리전에 Amazon Chime Voice Connector를 생성합니다.

```
aws chime create-voice-connector \
    --name newVoiceConnector \
    --aws-region us-west-2 \
    --require-encryption
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Chime 관리 안내서의 Amazon Chime Voice Connector 작업을 참조하세요.

- 자세한 API 내용은 명령 참조CreateVoiceConnector의 섹션을 참조하세요. AWS CLI

### `delete-account`

다음 코드 예시에서는 `delete-account`을 사용하는 방법을 보여 줍니다.

AWS CLI

계정을 삭제하려면

다음 `delete-account` 예제에서는 지정된 계정을 삭제합니다.

```
aws chime delete-account --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Chime 관리 안내서의 계정 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteAccount의 섹션을 참조하세요. AWS CLI

### `delete-phone-number`

다음 코드 예시에서는 `delete-phone-number`을 사용하는 방법을 보여 줍니다.

AWS CLI

전화번호를 삭제하려면

다음 `delete-phone-number` 예제에서는 지정된 전화번호를 삭제 대기열로 이동합니다.

```
aws chime delete-phone-number \
```

```
    --phone-number-id "+12065550100"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Chime 관리 안내서의 전화번호 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DeletePhoneNumber의 섹션을 참조하세요. AWS CLI

### delete-proxy-session

다음 코드 예시에서는 delete-proxy-session을 사용하는 방법을 보여 줍니다.

AWS CLI

프록시 세션을 삭제하려면

다음 delete-proxy-session 예제에서는 지정된 프록시 세션을 삭제합니다.

```
aws chime delete-proxy-session \
    --voice-connector-id abcdef1ghij2klmno3pqr4 \
    --proxy-session-id 123a4bc5-67d8-901e-2f3g-h4ghjk567891
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Chime 개발자 안내서의 프록시 전화 세션을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteProxySession의 섹션을 참조하세요. AWS CLI

### delete-room-membership

다음 코드 예시에서는 delete-room-membership을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자를 채팅방의 구성원으로 제거하려면

다음 delete-room-membership 예제에서는 지정된 채팅룸에서 지정된 멤버를 제거합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Chime 사용 설명서의 채팅룸 생성을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteRoomMembership의 섹션을 참조하세요. AWS CLI

### `delete-room`

다음 코드 예시에서는 `delete-room`을 사용하는 방법을 보여 줍니다.

AWS CLI

채팅방을 삭제하려면

다음 `delete-room` 예제에서는 지정된 채팅룸을 삭제하고 채팅룸 멤버십을 제거합니다.

```
aws chime delete-room \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Chime 사용 설명서의 채팅룸 생성을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteRoom의 섹션을 참조하세요. AWS CLI

### `delete-voice-connector-group`

다음 코드 예시에서는 `delete-voice-connector-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

title

다음 `delete-voice-connector-group` 예제에서는 지정된 Amazon Chime Voice Connector 그룹을 삭제합니다.

```
aws chime delete-voice-connector-group \
    --voice-connector-group-id 123a456b-c7d8-90e1-fg23-4h567jkl8901
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Chime 관리 안내서의 Amazon Chime Voice Connector 그룹 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteVoiceConnectorGroup의 섹션을 참조하세요. AWS CLI

### `delete-voice-connector-origination`

다음 코드 예시에서는 `delete-voice-connector-origination`을 사용하는 방법을 보여 줍니다.

AWS CLI

발신 설정을 삭제하려면

다음 `delete-voice-connector-origination` 예제에서는 지정된 Amazon Chime Voice Connector에서 오리진 호스트, 포트, 프로토콜, 우선 순위 및 가중치를 삭제합니다.

```
aws chime delete-voice-connector-origination \
    --voice-connector-id abcdef1ghij2klmno3pqr4
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Chime 관리 안내서의 Amazon Chime Voice Connector 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteVoiceConnectorOrigination의 섹션을 참조하세요. AWS CLI

### `delete-voice-connector-proxy`

다음 코드 예시에서는 `delete-voice-connector-proxy`을 사용하는 방법을 보여 줍니다.

AWS CLI

프록시 구성을 삭제하려면

다음 `delete-voice-connector-proxy` 예제에서는 Amazon Chime Voice Connector에서 프록시 구성을 삭제합니다.

```
aws chime delete-voice-connector-proxy \
    --voice-connector-id abcdef1ghij2klmno3pqr4
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Chime 개발자 안내서의 프록시 전화 세션을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteVoiceConnectorProxy의 섹션을 참조하세요. AWS CLI

### `delete-voice-connector-streaming-configuration`

다음 코드 예시에서는 `delete-voice-connector-streaming-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

스트리밍 구성을 삭제하려면

다음 `delete-voice-connector-streaming-configuration` 예제에서는 지정된 Amazon Chime Voice Connector에 대한 스트리밍 구성을 삭제합니다.

```
aws chime delete-voice-connector-streaming-configuration \
    --voice-connector-id abcdef1ghij2klmno3pqr4
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Chime 관리 안내서의 Kinesis에 Amazon Chime Voice Connector 데이터 스트리밍을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteVoiceConnectorStreamingConfiguration의 섹션을 참조하세요. AWS CLI

### `delete-voice-connector-termination-credentials`

다음 코드 예시에서는 `delete-voice-connector-termination-credentials`을 사용하는 방법을 보여 줍니다.

AWS CLI

종료 보안 인증 정보를 삭제하려면

다음 `delete-voice-connector-termination-credentials` 예제에서는 지정된 사용자 이름 및 Amazon Chime Voice Connector에 대한 종료 보안 인증 정보를 삭제합니다.

```
aws chime delete-voice-connector-termination-credentials \
```

```
    --voice-connector-id abcdef1ghij2klmno3pqr4 \
    --usernames "jdoe"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Chime 관리 안내서의 Amazon Chime Voice Connector 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteVoiceConnectorTerminationCredentials의 섹션을 참조하세요. AWS CLI

### delete-voice-connector-termination

다음 코드 예시에서는 delete-voice-connector-termination을 사용하는 방법을 보여 줍니다.

AWS CLI

종료 설정을 삭제하려면

다음 delete-voice-connector-termination 예제에서는 지정된 Amazon Chime Voice Connector에 대한 종료 설정을 삭제합니다.

```
aws chime delete-voice-connector-termination \
    --voice-connector-id abcdef1ghij2klmno3pqr4
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Chime 관리 안내서의 Amazon Chime Voice Connector 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteVoiceConnectorTermination의 섹션을 참조하세요. AWS CLI

### delete-voice-connector

다음 코드 예시에서는 delete-voice-connector을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon Chime Voice Connector를 삭제하려면

다음 delete-voice-connector 예제에서는

```
aws chime delete-voice-connector \
    --voice-connector-id abcdef1ghij2klmno3pqr4
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Chime 관리 안내서의 Amazon Chime Voice Connector 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteVoiceConnector의 섹션을 참조하세요. AWS CLI

### disassociate-phone-number-from-user

다음 코드 예시에서는 disassociate-phone-number-from-user을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자로부터 전화번호 연결을 해제하려면

다음 disassociate-phone-number-from-user 예제에서는 지정된 사용자의 전화번호를 연결 해제합니다.

```
aws chime disassociate-phone-number-from-user \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --user-id 1ab2345c-67de-8901-f23g-45h678901j2k
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Chime 관리 안내서의 사용자 전화번호 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DisassociatePhoneNumberFromUser의 섹션을 참조하세요. AWS CLI

### disassociate-phone-numbers-from-voice-connector-group

다음 코드 예시에서는 disassociate-phone-numbers-from-voice-connector-group을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon Chime Voice Connector 그룹에서 전화번호 연결을 해제하려면

다음 disassociate-phone-numbers-from-voice-connector-group 예제에서는 Amazon Chime Voice Connector 그룹에서 지정된 전화번호의 연결을 해제합니다.

```
aws chime disassociate-phone-numbers-from-voice-connector-group \
    --voice-connector-group-id 123a456b-c7d8-90e1-fg23-4h567jkl8901 \
    --e164-phone-numbers "+12065550100" "+12065550101"
```

출력:

```
{
    "PhoneNumberErrors": []
}
```

자세한 내용은 Amazon Chime 관리 안내서의 Amazon Chime Voice Connector 그룹 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DisassociatePhoneNumbersFromVoiceConnectorGroup의 섹션을 참조하세요. AWS CLI

### disassociate-phone-numbers-from-voice-connector

다음 코드 예시에서는 disassociate-phone-numbers-from-voice-connector을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon Chime Voice Connector에서 전화번호 연결을 해제하려면

다음 disassociate-phone-numbers-from-voice-connector 예제에서는 Amazon Chime Voice Connector에서 지정된 전화번호의 연결을 해제합니다.

```
aws chime disassociate-phone-numbers-from-voice-connector \
    --voice-connector-id abcdef1ghij2klmno3pqr4 \
    --e164-phone-numbers "+12065550100" "+12065550101"
```

출력:

```
{
    "PhoneNumberErrors": []
}
```

자세한 내용은 Amazon Chime 관리 안내서의 Amazon Chime Voice Connector 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DisassociatePhoneNumbersFromVoiceConnector의 섹션을 참조하세요. AWS CLI

### `disassociate-signin-delegate-groups-from-account`

다음 코드 예시에서는 `disassociate-signin-delegate-groups-from-account`을 사용하는 방법을 보여 줍니다.

AWS CLI

로그인 위임 그룹의 연결을 해제하려면

다음 `disassociate-signin-delegate-groups-from-account` 예제에서는 지정된 Amazon Chime 계정에서 지정된 로그인 위임 그룹의 연결을 해제합니다.

```
aws chime disassociate-signin-delegate-groups-from-account \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --group-names "my_users"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Chime 관리 안내서의 사용자 액세스 및 권한 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DisassociateSigninDelegateGroupsFromAccount의 섹션을 참조하세요. AWS CLI

### `get-account-settings`

다음 코드 예시에서는 `get-account-settings`을 사용하는 방법을 보여 줍니다.

AWS CLI

계정의 설정을 검색하려면

다음 `get-account-settings` 예제에서는 지정된 계정에 대한 계정 설정을 검색합니다.

```
aws chime get-account-settings --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

출력:

```
{
    "AccountSettings": {
        "DisableRemoteControl": false,
        "EnableDialOut": false
    }
}
```

자세한 내용은 Amazon Chime 관리 안내서의 Amazon Chime 계정 관리를 참조하세요.

- 자세한 API 내용은 명령 참조GetAccountSettings의 섹션을 참조하세요. AWS CLI

### get-account

다음 코드 예시에서는 get-account을 사용하는 방법을 보여 줍니다.

AWS CLI

계정의 세부 정보를 검색하려면

다음 get-account 예제에서는 지정된 Amazon Chime 계정에 대한 세부 정보를 검색합니다.

```
aws chime get-account \
    --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

출력:

```
{
    "Account": {
        "AwsAccountId": "111122223333",
        "AccountId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
        "Name": "EnterpriseDirectory",
        "AccountType": "EnterpriseDirectory",
        "CreatedTimestamp": "2018-12-20T18:38:02.181Z",
        "DefaultLicense": "Pro",
        "SupportedLicenses": [
            "Basic",
            "Pro"
        ],
        "SigninDelegateGroups": [
            {
```

```
                "GroupName": "myGroup"
            },
        ]
    }
}
```

자세한 내용은 Amazon Chime 관리 안내서의 Amazon Chime 계정 관리를 참조하세요.

- 자세한 API 내용은 명령 참조GetAccount의 섹션을 참조하세요. AWS CLI

### get-bot

다음 코드 예시에서는 get-bot을 사용하는 방법을 보여 줍니다.

AWS CLI

봇에 대한 세부 정보를 검색하려면

다음 get-bot 예제에서는 지정된 봇에 대한 세부 정보를 표시합니다.

```
aws chime get-bot \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --bot-id 123abcd4-5ef6-789g-0h12-34j56789012k
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Chime 개발자 안내서의 채팅 봇 업데이트를 참조하세요.

- 자세한 API 내용은 명령 참조GetBot의 섹션을 참조하세요. AWS CLI

### get-global-settings

다음 코드 예시에서는 get-global-settings을 사용하는 방법을 보여 줍니다.

AWS CLI

전역 설정을 가져오려면

다음 get-global-settings 예제에서는 관리자 AWS 계정과 연결된 Amazon Chime Business Calling 및 Amazon Chime Voice Connectors에 대한 통화 세부 정보 레코드를 저장하는 데 사용되는 S3 버킷 이름을 검색합니다.

```
aws chime get-global-settings
```

출력:

```
{
    "BusinessCalling": {
        "CdrBucket": "s3bucket"
    },
    "VoiceConnector": {
        "CdrBucket": "s3bucket"
    }
}
```

자세한 내용은 Amazon Chime 관리 안내서의 전역 설정 관리를 참조하세요.

- 자세한 API 내용은 명령 참조GetGlobalSettings의 섹션을 참조하세요. AWS CLI

### get-phone-number-order

다음 코드 예시에서는 get-phone-number-order을 사용하는 방법을 보여 줍니다.

AWS CLI

전화번호 주문에 대한 세부 정보를 가져오려면

다음 get-phone-number-order 예제에서는 지정된 전화번호 순서의 세부 정보를 표시합니다.

```
aws chime get-phone-number-order \
    --phone-number-order-id abc12345-de67-89f0-123g-h45i678j9012
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Chime 관리 안내서의 전화번호 작업을 참조하세요.

- 자세한 API 내용은 명령 참조GetPhoneNumberOrder의 섹션을 참조하세요. AWS CLI

### get-phone-number-settings

다음 코드 예시에서는 get-phone-number-settings을 사용하는 방법을 보여 줍니다.

AWS CLI

아웃바운드 통화 이름을 검색하려면

다음 get-phone-number-settings 예제에서는 호출 사용자 AWS 계정의 기본 아웃바운드 호출 이름을 검색합니다.

```
aws chime get-phone-number-settings
```

이 명령은 출력을 생성하지 않습니다. 출력:

```
{
    "CallingName": "myName",
    "CallingNameUpdatedTimestamp": "2019-10-28T18:56:42.911Z"
}
```

자세한 내용은 Amazon Chime 관리 안내서의 전화번호 작업을 참조하세요.

- 자세한 API 내용은 명령 참조GetPhoneNumberSettings의 섹션을 참조하세요. AWS CLI

### get-phone-number

다음 코드 예시에서는 get-phone-number을 사용하는 방법을 보여 줍니다.

AWS CLI

전화번호 세부 정보를 가져오려면

다음 get-phone-number 예제에서는 지정된 전화번호의 세부 정보를 표시합니다.

```
aws chime get-phone-number \
    --phone-number-id +12065550100
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Chime 관리 안내서의 전화번호 작업을 참조하세요.

- 자세한 API 내용은 명령 참조GetPhoneNumber의 섹션을 참조하세요. AWS CLI

### get-proxy-session

다음 코드 예시에서는 get-proxy-session을 사용하는 방법을 보여 줍니다.

AWS CLI

프록시 세션 세부 정보를 가져오려면

다음 get-proxy-session 예제에서는 지정된 프록시 세션의 세부 정보를 나열합니다.

```
aws chime get-proxy-session \
    --voice-connector-id abcdef1ghij2klmno3pqr4 \
    --proxy-session-id 123a4bc5-67d8-901e-2f3g-h4ghjk56789l
```

출력:

```
{
    "ProxySession": {
        "VoiceConnectorId": "abcdef1ghij2klmno3pqr4",
        "ProxySessionId": "123a4bc5-67d8-901e-2f3g-h4ghjk56789l",
        "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"
            }
        ]
    }
}
```

자세한 내용은 Amazon Chime 개발자 안내서의 프록시 전화 세션을 참조하세요.

- 자세한 API 내용은 명령 참조GetProxySession의 섹션을 참조하세요. AWS CLI

### get-room

다음 코드 예시에서는 get-room을 사용하는 방법을 보여 줍니다.

AWS CLI

채팅방에 대한 세부 정보를 가져오려면

다음 get-room 예제에서는 지정된 채팅룸에 대한 세부 정보를 표시합니다.

```
aws chime get-room \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j
```

출력:

```
{
    "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"
    }
```

```
}
```

자세한 내용은 Amazon Chime 사용 설명서의 채팅룸 생성을 참조하세요.

- 자세한 API 내용은 명령 참조GetRoom의 섹션을 참조하세요. AWS CLI

### get-user-settings

다음 코드 예시에서는 get-user-settings을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 설정을 검색하려면

다음 get-user-settings 예제에서는 지정된 사용자 설정을 표시합니다.

```
aws chime get-user-settings \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --user-id 1ab2345c-67de-8901-f23g-45h678901j2k
```

출력:

```
{
    "UserSettings": {
        "Telephony": {
            "InboundCalling": true,
            "OutboundCalling": true,
            "SMS": true
        }
    }
}
```

자세한 내용은 Amazon Chime 관리 안내서의 사용자 전화번호 관리를 참조하세요.

- 자세한 API 내용은 명령 참조GetUserSettings의 섹션을 참조하세요. AWS CLI

### get-user

다음 코드 예시에서는 get-user을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자에 대한 세부 정보를 가져오려면

다음 get-user 예제에서는 지정된 사용자의 세부 정보를 검색합니다.

```
aws chime get-user \
    --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \
    --user-id a1b2c3d4-5678-90ab-cdef-22222EXAMPLE
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Chime 관리 안내서의 사용자 관리를 참조하세요.

- 자세한 API 내용은 명령 참조GetUser의 섹션을 참조하세요. AWS CLI

### get-voice-connector-group

다음 코드 예시에서는 get-voice-connector-group을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon Chime Voice Connector 그룹에 대한 세부 정보를 가져오려면

다음 get-voice-connector-group 예제에서는 지정된 Amazon Chime Voice Connector 그룹에 대한 세부 정보를 표시합니다.

```
aws chime get-voice-connector-group \
```

```
    --voice-connector-group-id 123a456b-c7d8-90e1-fg23-4h567jkl8901
```

출력:

```
{
    "VoiceConnectorGroup": {
        "VoiceConnectorGroupId": "123a456b-c7d8-90e1-fg23-4h567jkl8901",
        "Name": "myGroup",
        "VoiceConnectorItems": [],
        "CreatedTimestamp": "2019-09-18T16:38:34.734Z",
        "UpdatedTimestamp": "2019-09-18T16:38:34.734Z"
    }
}
```

자세한 내용은 Amazon Chime 관리 안내서의 Amazon Chime Voice Connector 그룹 작업을 참조하세요.

- 자세한 API 내용은 명령 참조GetVoiceConnectorGroup의 섹션을 참조하세요. AWS CLI

### get-voice-connector-logging-configuration

다음 코드 예시에서는 get-voice-connector-logging-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

로깅 구성 세부 정보를 가져오려면

다음 get-voice-connector-logging-configuration 예제에서는 지정된 Amazon Chime Voice Connector에 대한 로깅 구성 세부 정보를 검색합니다.

```
aws chime get-voice-connector-logging-configuration \
    --voice-connector-id abcdef1ghij2klmno3pqr4
```

출력:

```
{
    "LoggingConfiguration": {
        "EnableSIPLogs": true
    }
```

```
}
```

자세한 내용은 Amazon Chime 관리 안내서의 Kinesis에 Amazon Chime Voice Connector Media 스트리밍을 참조하세요.

- 자세한 API 내용은 명령 참조GetVoiceConnectorLoggingConfiguration의 섹션을 참조하세요. AWS CLI

### `get-voice-connector-origination`

다음 코드 예시에서는 `get-voice-connector-origination`을 사용하는 방법을 보여 줍니다.

AWS CLI

오리진 설정을 검색하려면

다음 `get-voice-connector-origination` 예제에서는 지정된 Amazon Chime Voice Connector의 오리진 호스트, 포트, 프로토콜, 우선 순위 및 가중치를 검색합니다.

```
aws chime get-voice-connector-origination \
    --voice-connector-id abcdef1ghij2klmno3pqr4
```

출력:

```
{
    "Origination": {
        "Routes": [
            {
                "Host": "10.24.34.0",
                "Port": 1234,
                "Protocol": "TCP",
                "Priority": 1,
                "Weight": 5
            }
        ],
        "Disabled": false
    }
}
```

자세한 내용은 Amazon Chime 관리 안내서의 Amazon Chime Voice Connector 작업을 참조하세요.

- 자세한 API 내용은 명령 참조GetVoiceConnectorOrigination의 섹션을 참조하세요. AWS CLI

### `get-voice-connector-proxy`

다음 코드 예시에서는 `get-voice-connector-proxy`을 사용하는 방법을 보여 줍니다.

AWS CLI

프록시 구성 세부 정보를 가져오려면

다음 `get-voice-connector-proxy` 예제에서는 Amazon Chime Voice Connector의 프록시 구성 세부 정보를 가져옵니다.

```
aws chime get-voice-connector-proxy \
    --voice-connector-id abcdef1ghij2klmno3pqr4
```

출력:

```
{
    "Proxy": {
        "DefaultSessionExpiryMinutes": 60,
        "Disabled": false,
        "PhoneNumberCountries": [
            "US"
        ]
    }
}
```

자세한 내용은 Amazon Chime 개발자 안내서의 프록시 전화 세션을 참조하세요.

- 자세한 API 내용은 명령 참조GetVoiceConnectorProxy의 섹션을 참조하세요. AWS CLI

### `get-voice-connector-streaming-configuration`

다음 코드 예시에서는 `get-voice-connector-streaming-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

스트리밍 구성 세부 정보를 가져오려면

다음 `get-voice-connector-streaming-configuration` 예제에서는 지정된 Amazon Chime Voice Connector에 대한 스트리밍 구성 세부 정보를 가져옵니다.

```
aws chime get-voice-connector-streaming-configuration \
    --voice-connector-id abcdef1ghij2klmno3pqr4
```

출력:

```
{
    "StreamingConfiguration": {
        "DataRetentionInHours": 24,
        "Disabled": false
    }
}
```

자세한 내용은 Amazon Chime 관리 안내서의 Kinesis에 Amazon Chime Voice Connector 데이터 스트리밍을 참조하세요.

- 자세한 API 내용은 명령 참조GetVoiceConnectorStreamingConfiguration의 섹션을 참조하세요. AWS CLI

### `get-voice-connector-termination-health`

다음 코드 예시에서는 `get-voice-connector-termination-health`을 사용하는 방법을 보여 줍니다.

AWS CLI

종료 상태 세부 정보를 검색하려면

다음 `get-voice-connector-termination-health` 예제에서는 지정된 Amazon Chime Voice Connector에 대한 종료 상태 세부 정보를 검색합니다.

```
aws chime get-voice-connector-termination-health \
    --voice-connector-id abcdef1ghij2klmno3pqr4
```

출력:

```
{
    "TerminationHealth": {
        "Timestamp": "Fri Aug 23 16:45:55 UTC 2019",
```

```
        "Source": "10.24.34.0"
    }
}
```

자세한 내용은 Amazon Chime 관리 안내서의 Amazon Chime Voice Connector 작업을 참조하세요.

- 자세한 API 내용은 명령 참조GetVoiceConnectorTerminationHealth의 섹션을 참조하세요. AWS CLI

### get-voice-connector-termination

다음 코드 예시에서는 get-voice-connector-termination을 사용하는 방법을 보여 줍니다.

AWS CLI

종료 설정을 검색하려면

다음 get-voice-connector-termination 예제에서는 지정된 Amazon Chime Voice Connector에 대한 종료 설정을 검색합니다.

```
aws chime get-voice-connector-termination \
    --voice-connector-id abcdef1ghij2klmno3pqr4
```

이 명령은 출력을 생성하지 않습니다. 출력:

```
{
    "Termination": {
        "CpsLimit": 1,
        "DefaultPhoneNumber": "+12065550100",
        "CallingRegions": [
            "US"
        ],
        "CidrAllowedList": [
            "10.24.34.0/23"
        ],
        "Disabled": false
    }
}
```

자세한 내용은 Amazon Chime 관리 안내서의 Amazon Chime Voice Connector 작업을 참조하세요.

- 자세한 API 내용은 명령 참조GetVoiceConnectorTermination의 섹션을 참조하세요. AWS CLI

### `get-voice-connector`

다음 코드 예시에서는 `get-voice-connector`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon Chime Voice Connector에 대한 세부 정보를 가져오려면

다음 `get-voice-connector` 예제에서는 지정된 Amazon Chime Voice Connector의 세부 정보를 표시합니다.

```
aws chime get-voice-connector \
    --voice-connector-id abcdef1ghij2klmno3pqr4
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Chime 관리 안내서의 Amazon Chime Voice Connector 작업을 참조하세요.

- 자세한 API 내용은 명령 참조GetVoiceConnector의 섹션을 참조하세요. AWS CLI

### `invite-users`

다음 코드 예시에서는 `invite-users`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon Chime에 가입하도록 사용자를 초대하려면

다음 `invite-users` 예제에서는 지정된 Amazon Chime 계정에 사용자를 초대하는 이메일을 보냅니다.

```
aws chime invite-users \
    --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \
    --user-email-list "alejandror@example.com" "janed@example.com"
```

출력:

```
{
    "Invites": [
        {
            "InviteId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
            "Status": "Pending",
            "EmailAddress": "alejandror@example.com",
            "EmailStatus": "Sent"
        }
        {
            "InviteId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
            "Status": "Pending",
            "EmailAddress": "janed@example.com",
            "EmailStatus": "Sent"
        }
    ]
}
```

자세한 내용은 Amazon Chime 관리 안내서의 사용자 초대 및 일시 중지를 참조하세요.

- 자세한 API 내용은 명령 참조InviteUsers의 섹션을 참조하세요. AWS CLI

### `list-accounts`

다음 코드 예시에서는 `list-accounts`을 사용하는 방법을 보여 줍니다.

AWS CLI

계정 목록을 가져오려면

다음 `list-accounts` 예제에서는 관리자 계정의 Amazon Chime AWS 계정 목록을 검색합니다.

```
aws chime list-accounts
```

출력:

```
{
    "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"
                },
            ]
        }
    ]
}
```

자세한 내용은 Amazon Chime 관리 안내서의 Amazon Chime 계정 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ListAccounts의 섹션을 참조하세요. AWS CLI

### `list-bots`

다음 코드 예시에서는 `list-bots`을 사용하는 방법을 보여 줍니다.

AWS CLI

봇 목록을 검색하려면

다음 `list-bots` 예제에서는 지정된 Amazon Chime Enterprise 계정과 연결된 봇을 나열합니다.

```
aws chime list-bots \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Chime 개발자 안내서의 Amazon Chime에서 채팅 봇 사용을 참조하세요.

- 자세한 API 내용은 명령 참조ListBots의 섹션을 참조하세요. AWS CLI

### `list-phone-number-orders`

다음 코드 예시에서는 `list-phone-number-orders`을 사용하는 방법을 보여 줍니다.

AWS CLI

전화번호 주문을 나열하려면

다음 `list-phone-number-orders` 예제에서는 Amazon Chime 관리자 계정과 연결된 전화번호 주문을 나열합니다.

```
aws chime list-phone-number-orders
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Chime 관리 안내서의 전화번호 작업을 참조하세요.

- 자세한 API 내용은 명령 참조ListPhoneNumberOrders의 섹션을 참조하세요. AWS CLI

### `list-phone-numbers`

다음 코드 예시에서는 `list-phone-numbers`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon Chime 계정의 전화번호를 나열하려면

다음 `list-phone-numbers` 예제에서는 관리자의 Amazon Chime 계정과 연결된 전화번호를 나열합니다.

```
aws chime list-phone-numbers
```

이 명령은 출력을 생성하지 않습니다. 출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Chime 관리 안내서의 전화번호 작업을 참조하세요.

- 자세한 API 내용은 명령 참조ListPhoneNumbers의 섹션을 참조하세요. AWS CLI

### `list-proxy-sessions`

다음 코드 예시에서는 `list-proxy-sessions`을 사용하는 방법을 보여 줍니다.

AWS CLI

프록시 세션을 나열하려면

다음 `list-proxy-sessions` 예제에서는 Amazon Chime Voice Connector의 프록시 세션을 나열합니다.

```
aws chime list-proxy-sessions \
    --voice-connector-id abcdef1ghij2klmno3pqr4
```

출력:

```
{
    "ProxySession": {
        "VoiceConnectorId": "abcdef1ghij2klmno3pqr4",
        "ProxySessionId": "123a4bc5-67d8-901e-2f3g-h4ghjk56789l",
        "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"
            }
        ]
    }
}
```

자세한 내용은 Amazon Chime 개발자 안내서의 프록시 전화 세션을 참조하세요.

- 자세한 API 내용은 명령 참조ListProxySessions의 섹션을 참조하세요. AWS CLI

### `list-room-memberships`

다음 코드 예시에서는 `list-room-memberships`을 사용하는 방법을 보여 줍니다.

AWS CLI

룸 멤버십을 나열하려면

다음 `list-room-memberships` 예제에서는 지정된 채팅룸의 멤버십 세부 정보 목록을 표시합니다.

```
aws chime list-room-memberships \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Chime 사용 설명서의 채팅룸 생성을 참조하세요.

- 자세한 API 내용은 명령 참조ListRoomMemberships의 섹션을 참조하세요. AWS CLI

### `list-rooms`

다음 코드 예시에서는 `list-rooms`을 사용하는 방법을 보여 줍니다.

AWS CLI

채팅방을 나열하려면

다음 `list-rooms` 예제에서는 지정된 계정의 채팅방 목록을 표시합니다. 목록은 지정된 멤버가 속한 채팅방으로만 필터링됩니다.

```
aws chime list-rooms \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --member-id 1ab2345c-67de-8901-f23g-45h678901j2k
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Chime 사용 설명서의 채팅룸 생성을 참조하세요.

- 자세한 API 내용은 명령 참조ListRooms의 섹션을 참조하세요. AWS CLI

### list-users

다음 코드 예시에서는 list-users을 사용하는 방법을 보여 줍니다.

AWS CLI

계정의 사용자를 나열하려면

다음 list-users 예제에서는 지정된 Amazon Chime 계정의 사용자를 나열합니다.

```
aws chime list-users --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

출력:

```
{
    "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
            }
        }
    ]
}
```

자세한 내용은 Amazon Chime 관리 안내서의 사용자 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ListUsers의 섹션을 참조하세요. AWS CLI

### `list-voice-connector-groups`

다음 코드 예시에서는 `list-voice-connector-groups`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon Chime 계정에 대한 Amazon Chime Voice Connector 그룹을 나열하려면

다음 `list-voice-connector-groups` 예제에서는 관리자의 Amazon Chime 계정과 연결된 Amazon Chime Voice Connector 그룹을 나열합니다.

```
aws chime list-voice-connector-groups
```

출력:

```
{
    "VoiceConnectorGroups": [
        {
            "VoiceConnectorGroupId": "123a456b-c7d8-90e1-fg23-4h567jkl8901",
            "Name": "myGroup",
            "VoiceConnectorItems": [],
            "CreatedTimestamp": "2019-09-18T16:38:34.734Z",
            "UpdatedTimestamp": "2019-09-18T16:38:34.734Z"
        }
    ]
}
```

자세한 내용은 Amazon Chime 관리 안내서의 Amazon Chime Voice Connector 그룹 작업을 참조하세요.

- 자세한 API 내용은 명령 참조ListVoiceConnectorGroups의 섹션을 참조하세요. AWS CLI

### `list-voice-connector-termination-credentials`

다음 코드 예시에서는 `list-voice-connector-termination-credentials`을 사용하는 방법을 보여 줍니다.

AWS CLI

종료 자격 증명 목록을 검색하려면

다음 `list-voice-connector-termination-credentials` 예제에서는 지정된 Amazon Chime Voice Connector에 대한 종료 보안 인증 목록을 검색합니다.

```
aws chime list-voice-connector-termination-credentials \
    --voice-connector-id abcdef1ghij2klmno3pqr4
```

이 명령은 출력을 생성하지 않습니다. 출력:

```
{
    "Usernames": [
        "jdoe"
    ]
}
```

자세한 내용은 Amazon Chime 관리 안내서의 Amazon Chime Voice Connector 작업을 참조하세요.

- 자세한 API 내용은 명령 참조ListVoiceConnectorTerminationCredentials의 섹션을 참조하세요. AWS CLI

### `list-voice-connectors`

다음 코드 예시에서는 `list-voice-connectors`을 사용하는 방법을 보여 줍니다.

AWS CLI

계정의 Amazon Chime Voice Connector를 나열하려면

다음 `list-voice-connectors` 예제에서는 발신자의 계정과 연결된 Amazon Chime Voice Connector를 나열합니다.

```
aws chime list-voice-connectors
```

출력:

```
{
    "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"
        }
    ]
```

```
}
```

자세한 내용은 Amazon Chime 관리 안내서의 Amazon Chime Voice Connector 작업을 참조하세요.

- 자세한 API 내용은 명령 참조ListVoiceConnectors의 섹션을 참조하세요. AWS CLI

### `logout-user`

다음 코드 예시에서는 `logout-user`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자를 로그아웃하려면

다음 `logout-user` 예제에서는 지정된 사용자를 로그아웃합니다.

```
aws chime logout-user \
    --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \
    --user-id a1b2c3d4-5678-90ab-cdef-22222EXAMPLE
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조LogoutUser의 섹션을 참조하세요. AWS CLI

### `put-voice-connector-logging-configuration`

다음 코드 예시에서는 `put-voice-connector-logging-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon Chime Voice Connector에 대한 로깅 구성을 추가하려면

다음 `put-voice-connector-logging-configuration` 예제에서는 지정된 Amazon Chime Voice Connector에 대한 SIP 로깅 구성을 활성화합니다.

```
aws chime put-voice-connector-logging-configuration \
    --voice-connector-id abcdef1ghij2klmno3pqr4 \
    --logging-configuration EnableSIPLogs=true
```

출력:

```
{
    "LoggingConfiguration": {
        "EnableSIPLogs": true
    }
}
```

자세한 내용은 Amazon Chime 관리 안내서의 Kinesis에 Amazon Chime Voice Connector Media 스트리밍을 참조하세요.

- 자세한 API 내용은 명령 참조PutVoiceConnectorLoggingConfiguration의 섹션을 참조하세요. AWS CLI

### put-voice-connector-origination

다음 코드 예시에서는 put-voice-connector-origination을 사용하는 방법을 보여 줍니다.

AWS CLI

발신 설정을 설정하려면

다음 put-voice-connector-origination 예제에서는 지정된 Amazon Chime Voice Connector의 오리진 호스트, 포트, 프로토콜, 우선 순위 및 가중치를 설정합니다.

```
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
```

출력:

```
{
    "Origination": {
        "Routes": [
            {
                "Host": "10.24.34.0",
                "Port": 1234,
                "Protocol": "TCP",
                "Priority": 1,
                "Weight": 5
```

```
            }
        ],
        "Disabled": false
    }
}
```

자세한 내용은 Amazon Chime 관리 안내서의 Amazon Chime Voice Connector 작업을 참조하세요.

- 자세한 API 내용은 명령 참조PutVoiceConnectorOrigination의 섹션을 참조하세요. AWS CLI

### put-voice-connector-proxy

다음 코드 예시에서는 put-voice-connector-proxy을 사용하는 방법을 보여 줍니다.

AWS CLI

프록시 구성을 배치하려면

다음 put-voice-connector-proxy 예제에서는 프록시 구성을 Amazon Chime Voice Connector로 설정합니다.

```
aws chime put-voice-connector-proxy \
    --voice-connector-id abcdef1ghij2klmno3pqr4 \
    --default-session-expiry-minutes 60 \
    --phone-number-pool-countries "US"
```

출력:

```
{
    "Proxy": {
        "DefaultSessionExpiryMinutes": 60,
        "Disabled": false,
        "PhoneNumberCountries": [
            "US"
        ]
    }
}
```

자세한 내용은 Amazon Chime 개발자 안내서의 프록시 전화 세션을 참조하세요.

- 자세한 API 내용은 명령 참조PutVoiceConnectorProxy의 섹션을 참조하세요. AWS CLI

### put-voice-connector-streaming-configuration

다음 코드 예시에서는 put-voice-connector-streaming-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

스트리밍 구성을 생성하려면

다음 put-voice-connector-streaming-configuration 예제에서는 지정된 Amazon Chime Voice Connector에 대한 스트리밍 구성을 생성합니다. Amazon Chime Voice Connector에서 Amazon Kinesis 로 미디어 스트리밍을 활성화하고 데이터 보존 기간을 24시간으로 설정합니다.

```
aws chime put-voice-connector-streaming-configuration \
    --voice-connector-id abcdef1ghij2klmno3pqr4 \
    --streaming-configuration DataRetentionInHours=24,Disabled=false
```

출력:

```
{
    "StreamingConfiguration": {
        "DataRetentionInHours": 24,
        "Disabled": false
    }
}
```

자세한 내용은 Amazon Chime 관리 안내서의 Kinesis에 Amazon Chime Voice Connector 데이터 스트리밍을 참조하세요.

- 자세한 API 내용은 명령 참조PutVoiceConnectorStreamingConfiguration의 섹션을 참조하세요. AWS CLI

### put-voice-connector-termination-credentials

다음 코드 예시에서는 put-voice-connector-termination-credentials을 사용하는 방법을 보여 줍니다.

AWS CLI

종료 자격 증명을 설정하려면

다음 put-voice-connector-termination-credentials 예제에서는 지정된 Amazon Chime Voice Connector에 대한 종료 보안 인증 정보를 설정합니다.

```
aws chime put-voice-connector-termination-credentials \
    --voice-connector-id abcdef1ghij2klmno3pqr4 \
    --credentials Username="jdoe",Password="XXXXXXXX"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Chime 관리 안내서의 Amazon Chime Voice Connector 작업을 참조하세요.

- 자세한 API 내용은 명령 참조PutVoiceConnectorTerminationCredentials의 섹션을 참조하세요. AWS CLI

### put-voice-connector-termination

다음 코드 예시에서는 put-voice-connector-termination을 사용하는 방법을 보여 줍니다.

AWS CLI

종료 설정을 설정하려면

다음 put-voice-connector-termination 예제에서는 지정된 Amazon Chime Voice Connector에 대해 호출 리전과 허용된 IP 호스트 종료 설정을 설정합니다.

```
aws chime put-voice-connector-termination \
    --voice-connector-id abcdef1ghij2klmno3pqr4 \
    --termination CallingRegions="US",CidrAllowedList="10.24.34.0/23",Disabled=false
```

출력:

```
{
    "Termination": {
        "CpsLimit": 0,
        "CallingRegions": [
            "US"
        ],
        "CidrAllowedList": [
            "10.24.34.0/23"
        ],
```

```
        "Disabled": false
    }
}
```

자세한 내용은 Amazon Chime 관리 안내서의 Amazon Chime Voice Connector 작업을 참조하세요.

- 자세한 API 내용은 명령 참조PutVoiceConnectorTermination의 섹션을 참조하세요. AWS CLI

### regenerate-security-token

다음 코드 예시에서는 regenerate-security-token을 사용하는 방법을 보여 줍니다.

AWS CLI

보안 토큰을 재생성하려면

다음 regenerate-security-token 예제에서는 지정된 봇에 대한 보안 토큰을 재생성합니다.

```
aws chime regenerate-security-token \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --bot-id 123abcd4-5ef6-789g-0h12-34j56789012k
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Chime 개발자 안내서의 채팅 봇 요청 인증을 참조하세요.

- 자세한 API 내용은 명령 참조RegenerateSecurityToken의 섹션을 참조하세요. AWS CLI

### reset-personal-pin

다음 코드 예시에서는 reset-personal-pin을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자의 개인 회의를 재설정하려면 PIN

다음 reset-personal-pin 예제에서는 지정된 사용자의 개인 회의 를 재설정합니다PIN.

```
aws chime reset-personal-pin \
    --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
    --user-id a1b2c3d4-5678-90ab-cdef-22222EXAMPLE
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Chime 관리 안내서의 개인 회의 변경을 PINs 참조하세요.

- 자세한 API 내용은 명령 참조ResetPersonalPin의 섹션을 참조하세요. AWS CLI

### restore-phone-number

다음 코드 예시에서는 restore-phone-number을 사용하는 방법을 보여 줍니다.

AWS CLI

전화번호를 복원하려면

다음 `restore-phone-number` 예제에서는 삭제 대기열에서 지정된 전화번호를 복원합니다.

```
aws chime restore-phone-number \
    --phone-number-id "+12065550100"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Chime 관리 안내서의 전화번호 작업을 참조하세요.

- 자세한 API 내용은 명령 참조RestorePhoneNumber의 섹션을 참조하세요. AWS CLI

### `search-available-phone-numbers`

다음 코드 예시에서는 `search-available-phone-numbers`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 전화번호를 검색하려면

다음 search-available-phone-numbers 예제에서는 지역 코드별로 사용 가능한 전화번호를 검색합니다.

```
aws chime search-available-phone-numbers \
    --area-code "206"
```

출력:

```
{
    "E164PhoneNumbers": [
        "+12065550100",
        "+12065550101",
        "+12065550102",
        "+12065550103",
        "+12065550104",
        "+12065550105",
        "+12065550106",
        "+12065550107",
        "+12065550108",
        "+12065550109",
    ]
}
```

자세한 내용은 Amazon Chime 관리 안내서의 전화번호 작업을 참조하세요.

- 자세한 API 내용은 명령 참조SearchAvailablePhoneNumbers의 섹션을 참조하세요. AWS CLI

### update-account-settings

다음 코드 예시에서는 update-account-settings을 사용하는 방법을 보여 줍니다.

AWS CLI

계정의 설정을 업데이트하려면

다음 update-account-settings 예제에서는 지정된 Amazon Chime 계정에 대한 공유 화면의 원격 제어를 비활성화합니다.

```
aws chime update-account-settings \
    --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \
    --account-settings DisableRemoteControl=true
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조UpdateAccountSettings의 섹션을 참조하세요. AWS CLI

### update-account

다음 코드 예시에서는 update-account을 사용하는 방법을 보여 줍니다.

AWS CLI

계정을 업데이트하려면

다음 update-account 예제에서는 지정된 계정 이름을 업데이트합니다.

```
aws chime update-account \
    --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \
    --name MyAccountName
```

출력:

```
{
    "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"
            },
        ]
    }
}
```

자세한 내용은 Amazon Chime 관리 안내서의 계정 이름 바꾸기를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateAccount의 섹션을 참조하세요. AWS CLI

### update-bot

다음 코드 예시에서는 update-bot을 사용하는 방법을 보여 줍니다.

AWS CLI

봇을 업데이트하려면

다음 update-bot 예제에서는 지정된 봇의 상태를 업데이트하여 실행을 중지합니다.

```
aws chime update-bot \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --bot-id 123abcd4-5ef6-789g-0h12-34j56789012k \
    --disabled
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Chime 개발자 안내서의 채팅 봇 업데이트를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateBot의 섹션을 참조하세요. AWS CLI

### update-global-settings

다음 코드 예시에서는 update-global-settings을 사용하는 방법을 보여 줍니다.

AWS CLI

전역 설정을 업데이트하려면

다음 update-global-settings 예제에서는 관리자 AWS 계정과 연결된 Amazon Chime Business Calling 및 Amazon Chime Voice Connectors에 대한 통화 세부 정보 레코드를 저장하는 데 사용되는 S3 버킷을 업데이트합니다.

```
aws chime update-global-settings \
    --business-calling CdrBucket="s3bucket" \
    --voice-connector CdrBucket="s3bucket"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Chime 관리 안내서의 전역 설정 관리를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateGlobalSettings의 섹션을 참조하세요. AWS CLI

### update-phone-number-settings

다음 코드 예시에서는 update-phone-number-settings을 사용하는 방법을 보여 줍니다.

AWS CLI

아웃바운드 통화 이름을 업데이트하려면

다음 update-phone-number-settings 예제에서는 관리자 AWS 계정의 기본 아웃바운드 통화 이름을 업데이트합니다.

```
aws chime update-phone-number-settings \
    --calling-name "myName"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Chime 관리 안내서의 전화번호 작업을 참조하세요.

- 자세한 API 내용은 명령 참조UpdatePhoneNumberSettings의 섹션을 참조하세요. AWS CLI

### update-phone-number

다음 코드 예시에서는 update-phone-number을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 전화번호의 제품 유형을 업데이트하려면

다음 update-phone-number 예제에서는 지정된 전화번호의 제품 유형을 업데이트합니다.

```
aws chime update-phone-number \
    --phone-number-id "+12065550100" \
    --product-type "BusinessCalling"
```

출력:

```
{
    "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"
    }
}
```

예제 2: 전화번호의 아웃바운드 통화 이름을 업데이트하려면

다음 update-phone-number 예제에서는 지정된 전화번호의 아웃바운드 통화 이름을 업데이트합니다.

aws 차임 update-phone-number --phone-number-id '+12065550100' --calling-name 'phonenumber2'

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Chime 관리 안내서의 전화번호 작업을 참조하세요.

- 자세한 API 내용은 명령 참조UpdatePhoneNumber의 섹션을 참조하세요. AWS CLI

### update-proxy-session

다음 코드 예시에서는 update-proxy-session을 사용하는 방법을 보여 줍니다.

AWS CLI

프록시 세션을 업데이트하려면

다음 update-proxy-session 예제에서는 프록시 세션 기능을 업데이트합니다.

```
aws chime update-proxy-session \
    --voice-connector-id abcdef1ghij2klmno3pqr4 \
    --proxy-session-id 123a4bc5-67d8-901e-2f3g-h4ghjk56789l \
    --capabilities "Voice"
```

출력:

```
{
```

```
    "ProxySession": {
        "VoiceConnectorId": "abcdef1ghij2klmno3pqr4",
        "ProxySessionId": "123a4bc5-67d8-901e-2f3g-h4ghjk56789l",
        "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"
            }
        ]
    }
}
```

자세한 내용은 Amazon Chime 개발자 안내서의 프록시 전화 세션을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateProxySession의 섹션을 참조하세요. AWS CLI

### update-room-membership

다음 코드 예시에서는 update-room-membership을 사용하는 방법을 보여 줍니다.

AWS CLI

룸 멤버십을 업데이트하려면

다음 update-room-membership 예제에서는 지정된 채팅룸 멤버의 역할을 로 수정합니다Administrator.

```
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
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Chime 사용 설명서의 채팅룸 생성을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateRoomMembership의 섹션을 참조하세요. AWS CLI

### update-room

다음 코드 예시에서는 update-room을 사용하는 방법을 보여 줍니다.

AWS CLI

채팅방을 업데이트하려면

다음 update-room 예제에서는 지정된 채팅룸의 이름을 수정합니다.

```
aws chime update-room \
    --account-id 12a3456b-7c89-012d-3456-78901e23fg45 \
    --room-id abcd1e2d-3e45-6789-01f2-3g45h67i890j \
    --name teamRoom
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Chime 사용 설명서의 채팅룸 생성을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateRoom의 섹션을 참조하세요. AWS CLI

### update-user-settings

다음 코드 예시에서는 update-user-settings을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 설정을 업데이트하려면

다음 update-user-settings 예제에서는 지정된 사용자가 인바운드 및 아웃바운드 전화를 걸고 SMS 메시지를 보내고 받을 수 있습니다.

```
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}"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Chime 관리 안내서의 사용자 전화번호 관리를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateUserSettings의 섹션을 참조하세요. AWS CLI

### update-user

다음 코드 예시에서는 update-user을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 세부 정보를 업데이트하려면

이 예제에서는 지정된 사용자에 대해 지정된 세부 정보를 업데이트합니다.

명령:

```
aws chime update-user \
    --account-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \
    --user-id a1b2c3d4-5678-90ab-cdef-22222EXAMPLE \
    --license-type "Basic"
```

출력:

```
{
    "User": {
        "UserId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE"
    }
}
```

- 자세한 API 내용은 명령 참조UpdateUser의 섹션을 참조하세요. AWS CLI

### update-voice-connector-group

다음 코드 예시에서는 update-voice-connector-group을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon Chime Voice Connector 그룹의 세부 정보를 업데이트하려면

다음 update-voice-connector-group 예제에서는 지정된 Amazon Chime Voice Connector 그룹의 세부 정보를 업데이트합니다.

```
aws chime update-voice-connector-group \
    --voice-connector-group-id 123a456b-c7d8-90e1-fg23-4h567jkl8901 \
    --name "newGroupName" \
    --voice-connector-items VoiceConnectorId=abcdef1ghij2klmno3pqr4,Priority=1
```

출력:

```
{
    "VoiceConnectorGroup": {
        "VoiceConnectorGroupId": "123a456b-c7d8-90e1-fg23-4h567jkl8901",
        "Name": "newGroupName",
        "VoiceConnectorItems": [
            {
```

```
                "VoiceConnectorId": "abcdef1ghij2klmno3pqr4",
                "Priority": 1
            }
        ],
        "CreatedTimestamp": "2019-09-18T16:38:34.734Z",
        "UpdatedTimestamp": "2019-10-28T19:00:57.081Z"
    }
}
```

자세한 내용은 Amazon Chime 관리 안내서의 Amazon Chime Voice Connector 그룹 작업을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateVoiceConnectorGroup의 섹션을 참조하세요. AWS CLI

### update-voice-connector

다음 코드 예시에서는 update-voice-connector을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon Chime Voice Connector에 대한 세부 정보를 업데이트하려면

다음 update-voice-connector 예제에서는 지정된 Amazon Chime Voice Connector의 이름을 업데이트합니다.

```
aws chime update-voice-connector \
    --voice-connector-id abcdef1ghij2klmno3pqr4 \
    --name newName \
    --require-encryption
```

출력:

```
{
    "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"
    }
```

```
}
```

자세한 내용은 Amazon Chime 관리 안내서의 Amazon Chime Voice Connector 작업을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateVoiceConnector의 섹션을 참조하세요. AWS CLI

## 를 사용한 Cloud Control API 예제 AWS CLI

다음 코드 예제에서는 Cloud Control과 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다API.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `create-resource`

다음 코드 예시에서는 `create-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스를 생성하려면

다음 `create-resource` 예제에서는 ResourceExample보존 기간이 168시간이고 샤드 수가 3인 : AWS:Kinesis::Stream 리소스를 생성합니다.

```
aws cloudcontrol create-resource \
    --type-name AWS::Kinesis::Stream \
    --desired-state "{\"Name\": \"ResourceExample\",\"RetentionPeriodHours\":168,
 \"ShardCount\":3}"
```

출력:

```
{
    "ProgressEvent": {
        "EventTime": 1632506656.706,
        "TypeName": "AWS::Kinesis::Stream",
        "OperationStatus": "IN_PROGRESS",
        "Operation": "CREATE",
        "Identifier": "ResourceExample",
        "RequestToken": "20999d87-e304-4725-ad84-832dcbfd7fc5"
    }
}
```

자세한 내용은 Cloud Control API 사용 설명서의 리소스 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateResource의 섹션을 참조하세요. AWS CLI

### `delete-resource`

다음 코드 예시에서는 `delete-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스를 삭제하려면

다음 `delete-resource` 예제에서는 AWS 계정 ResourceExample 에서 식별자가 포함된 AWS::Kinesis::Stream 리소스를 삭제합니다.

```
aws cloudcontrol delete-resource \
    --type-name AWS::Kinesis::Stream \
    --identifier ResourceExample
```

출력:

```
{
    "ProgressEvent": {
        "TypeName": "AWS::Kinesis::Stream",
        "Identifier": "ResourceExample",
        "RequestToken": "e48f26ff-d0f9-4ab8-a878-120db1edf111",
        "Operation": "DELETE",
        "OperationStatus": "IN_PROGRESS",
        "EventTime": 1632950300.14
    }
}
```

자세한 내용은 Cloud Control API 사용 설명서의 리소스 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteResource의 섹션을 참조하세요. AWS CLI

### `get-resource-request-status`

다음 코드 예시에서는 `get-resource-request-status`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 요청의 상태 정보를 가져오려면

다음 `get-resource-request-status` 예제에서는 지정된 리소스 요청에 대한 상태 정보를 반환합니다.

```
aws cloudcontrol get-resource-request-status \
    --request-token "e1a6b86e-46bd-41ac-bfba-001234567890"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Cloud Control API 사용 설명서의 리소스 작업 요청 관리를 참조하세요.

- 자세한 API 내용은 명령 참조GetResourceRequestStatus의 섹션을 참조하세요. AWS CLI

### `get-resource`

다음 코드 예시에서는 `get-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스의 현재 상태를 가져오려면

다음 `get-resource` 예제에서는 AWS::Kinesis::Stream 리소스의 현재 상태를 반환합니다 ResourceExample.

```
aws cloudcontrol get-resource \
    --type-name AWS::Kinesis::Stream \
    --identifier ResourceExample
```

출력:

```
{
    "TypeName": "AWS::Kinesis::Stream",
    "ResourceDescription": {
        "Identifier": "ResourceExample",
        "Properties": "{\"Arn\":\"arn:aws:kinesis:us-west-2:099908667365:stream/
ResourceExample\",\"RetentionPeriodHours\":168,\"Name\":\"ResourceExample\",
\"ShardCount\":3}"
    }
}
```

자세한 내용은 Cloud Control API 사용 설명서의 리소스의 현재 상태 읽기를 참조하세요.

- 자세한 API 내용은 명령 참조GetResource의 섹션을 참조하세요. AWS CLI

### `list-resource-requests`

다음 코드 예시에서는 `list-resource-requests`을 사용하는 방법을 보여 줍니다.

AWS CLI

활성 리소스 작업 요청을 나열하려면

다음 `list-resource-requests` 예제에서는 에 대한 리소스 요청CREATE과 AWS 계정에서 실패한 UPDATE 작업을 나열합니다.

```
aws cloudcontrol list-resource-requests \
    --resource-request-status-filter Operations=CREATE,OperationStatuses=FAILED
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Cloud Control API 사용 설명서의 리소스 작업 요청 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ListResourceRequests의 섹션을 참조하세요. AWS CLI

### `list-resources`

다음 코드 예시에서는 `list-resources`을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 유형의 리소스를 나열하려면

다음 `list-resources` 예제에서는 AWS 계정에 프로비저닝된 AWS::Kinesis::Stream 리소스를 나열합니다.

```
aws cloudcontrol list-resources \
    --type-name AWS::Kinesis::Stream
```

출력:

```
{
    "TypeName": "AWS::Kinesis::Stream",
    "ResourceDescriptions": [
        {
            "Identifier": "MyKinesisStream",
```

```
            "Properties": "{\"Name\":\"MyKinesisStream\"}"
        },
        {
            "Identifier": "AnotherStream",
            "Properties": "{\"Name\":\"AnotherStream\"}"
        }
    ]
}
```

자세한 내용은 Cloud Control API 사용 설명서의 리소스 검색을 참조하세요.

- 자세한 API 내용은 명령 참조ListResources의 섹션을 참조하세요. AWS CLI

### update-resource

다음 코드 예시에서는 update-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

기존 리소스의 속성을 업데이트하려면

다음 update-resource 예제에서는 이름이 인 AWS::Logs::LogGroup resource의 보존 정책을 ExampleLogGroup 90일로 업데이트합니다.

```
aws cloudcontrol update-resource \
    --type-name AWS::Logs::LogGroup \
    --identifier ExampleLogGroup \
    --patch-document "[{\"op\":\"replace\",\"path\":\"/RetentionInDays\",\"value
\":90}]"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Cloud Control API 사용 설명서의 리소스 업데이트를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateResource의 섹션을 참조하세요. AWS CLI

## AWS Cloud Map 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS Cloud Map.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `create-private-dns-namespace`

다음 코드 예시에서는 `create-private-dns-namespace`을 사용하는 방법을 보여 줍니다.

AWS CLI

프라이빗 DNS 네임스페이스를 생성하려면

다음 `create-private-dns-namespace` 예제에서는 프라이빗 DNS 네임스페이스를 생성합니다.

```
aws servicediscovery create-private-dns-namespace \
    --name example.com \
    --vpc vpc-1c56417b
```

출력:

```
{
    "OperationId": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd"
```

```
}
```

작업이 성공했는지 확인하기 위해 를 실행할 수 있습니다`get-operation`. 자세한 내용은 get-operation 을 참조하세요.

자세한 내용은 AWS Cloud Map 개발자 안내서의 네임스페이스 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreatePrivateDnsNamespace의 섹션을 참조하세요. AWS CLI

### `create-service`

다음 코드 예시에서는 `create-service`을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스를 생성하려면

다음 `create-service` 예제에서는 서비스를 생성합니다.

```
aws servicediscovery create-service \
    --name myservice \
    --namespace-id  ns-ylexjili4cdxy3xm \
    --dns-config "NamespaceId=ns-
ylexjili4cdxy3xm,RoutingPolicy=MULTIVALUE,DnsRecords=[{Type=A,TTL=60}]"
```

출력:

```
{
        "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"
    }
}
```

자세한 내용은 AWS Cloud Map 개발자 안내서의 서비스 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateService의 섹션을 참조하세요. AWS CLI

### delete-namespace

다음 코드 예시에서는 delete-namespace을 사용하는 방법을 보여 줍니다.

AWS CLI

네임스페이스를 삭제하려면

다음 delete-namespace 예제에서는 네임스페이스를 삭제합니다.

```
aws servicediscovery delete-namespace \
    --id ns-ylexjili4cdxy3xm
```

출력:

```
{
    "OperationId": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k98y6drk"
}
```

작업이 성공했는지 확인하기 위해 를 실행할 수 있습니다get-operation. 자세한 내용은 get-operation 을 참조하세요.

자세한 내용은 AWS Cloud Map 개발자 안내서의 네임스페이스 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteNamespace의 섹션을 참조하세요. AWS CLI

### delete-service

다음 코드 예시에서는 delete-service을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스를 삭제하는 방법

다음 delete-service 예제에서는 서비스를 삭제합니다.

```
aws servicediscovery delete-service \
    --id srv-p5zdwlg5uvvzjita
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Cloud Map 개발자 안내서의 서비스 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteService의 섹션을 참조하세요. AWS CLI

### deregister-instance

다음 코드 예시에서는 deregister-instance을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스 인스턴스 등록을 취소하려면

다음 deregister-instance 예제에서는 서비스 인스턴스의 등록을 취소합니다.

```
aws servicediscovery deregister-instance \
    --service-id srv-p5zdwlg5uvvzjita \
    --instance-id myservice-53
```

출력:

```
{
    "OperationId": "4yejorelbukcjzpnr6tlmrghsjwpngf4-k98rnaiq"
}
```

작업이 성공했는지 확인하기 위해 를 실행할 수 있습니다get-operation. 자세한 내용은 get-operation 을 참조하세요.

자세한 내용은 AWS Cloud Map 개발자 안내서의 서비스 인스턴스 등록 취소를 참조하세요.

- 자세한 API 내용은 명령 참조DeregisterInstance의 섹션을 참조하세요. AWS CLI

### discover-instances

다음 코드 예시에서는 discover-instances을 사용하는 방법을 보여 줍니다.

AWS CLI

등록된 인스턴스를 검색하려면

다음 discover-instances 예제에서는 등록된 인스턴스를 검색합니다.

```
aws servicediscovery discover-instances \
    --namespace-name example.com \
    --service-name myservice \
    --max-results 10 \
    --health-status ALL
```

출력:

```
{
    "Instances": [
        {
            "InstanceId": "myservice-53",
            "NamespaceName": "example.com",
            "ServiceName": "myservice",
            "HealthStatus": "UNKNOWN",
            "Attributes": {
                "AWS_INSTANCE_IPV4": "172.2.1.3",
                "AWS_INSTANCE_PORT": "808"
            }
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DiscoverInstances의 섹션을 참조하세요. AWS CLI

### get-operation

다음 코드 예시에서는 get-operation을 사용하는 방법을 보여 줍니다.

AWS CLI

작업 결과를 가져오려면

다음 get-operation 예제는 작업의 결과를 가져옵니다.

```
aws servicediscovery get-operation \
```

```
    --operation-id gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd
```

출력:

```
{
    "Operation": {
        "Id": "gv4g5meo7ndmeh4fqskygvk23d2fijwa-k9302yzd",
        "Type": "CREATE_NAMESPACE",
        "Status": "SUCCESS",
        "CreateDate": 1587055860.121,
        "UpdateDate": 1587055900.469,
        "Targets": {
            "NAMESPACE": "ns-ylexjili4cdxy3xm"
        }
    }
}
```

- 자세한 API 내용은 명령 참조GetOperation의 섹션을 참조하세요. AWS CLI

### `list-instances`

다음 코드 예시에서는 `list-instances`을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스 인스턴스를 나열하려면

다음 `list-instances` 예제에서는 서비스 인스턴스를 나열합니다.

```
aws servicediscovery list-instances \
    --service-id srv-qzpwvt2tfqcegapy
```

출력:

```
{
    "Instances": [
        {
            "Id": "i-06bdabbae60f65a4e",
            "Attributes": {
                "AWS_INSTANCE_IPV4": "172.2.1.3",
                "AWS_INSTANCE_PORT": "808"
            }
```

```
        }
    ]
}
```

자세한 내용은 AWS Cloud Map 개발자 안내서의 서비스 인스턴스 목록 보기를 참조하세요.

- 자세한 API 내용은 명령 참조ListInstances의 섹션을 참조하세요. AWS CLI

### `list-namespaces`

다음 코드 예시에서는 `list-namespaces`을 사용하는 방법을 보여 줍니다.

AWS CLI

네임스페이스를 나열하려면

다음 `list-namespaces` 예제에서는 네임스페이스를 나열합니다.

```
aws servicediscovery list-namespaces
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS Cloud Map 개발자 안내서의 네임스페이스 목록 보기를 참조하세요.

- 자세한 API 내용은 명령 참조ListNamespaces의 섹션을 참조하세요. AWS CLI

### `list-services`

다음 코드 예시에서는 `list-services`을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스를 나열하려면

다음 `list-services` 예제에서는 서비스를 나열합니다.

```
aws servicediscovery list-services
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 AWS Cloud Map 개발자 안내서의 서비스 목록 보기를 참조하세요.

- 자세한 API 내용은 명령 참조ListServices의 섹션을 참조하세요. AWS CLI

### register-instance

다음 코드 예시에서는 register-instance을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스 인스턴스를 등록하려면

다음 register-instance 예제에서는 서비스 인스턴스를 등록합니다.

```
aws servicediscovery register-instance \
    --service-id srv-p5zdwlg5uvvzjita \
    --instance-id myservice-53 \
```

```
    --attributes=AWS_INSTANCE_IPV4=172.2.1.3,AWS_INSTANCE_PORT=808
```

출력:

```
{
    "OperationId": "4yejorelbukcjzpnr6tlmrghsjwpngf4-k95yg2u7"
}
```

작업이 성공했는지 확인하기 위해 를 실행할 수 있습니다`get-operation`. 자세한 내용은 get-operation 을 참조하세요.

자세한 내용은 AWS Cloud Map 개발자 안내서의 인스턴스 등록을 참조하세요.

- 자세한 API 내용은 명령 참조RegisterInstance의 섹션을 참조하세요. AWS CLI

## AWS Cloud9 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS Cloud9.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `create-environment-ec2`

다음 코드 예시에서는 `create-environment-ec2`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS Cloud9 EC2 개발 환경을 생성하려면

다음 `create-environment-ec2` 예제에서는 지정된 설정을 사용하여 AWS Cloud9 개발 환경을 생성하고 Amazon Elastic Compute Cloud(AmazonEC2) 인스턴스를 시작한 다음 인스턴스에서 환경으로 연결합니다.

```
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
```

출력:

```
{
    "environmentId": "8a34f51ce1e04a08882f1e811bd706EX"
}
```

자세한 내용은 AWS Cloud9 사용 설명서의 EC2 환경 생성을 참조하세요.

- API 자세한 내용은 AWS CLI 명령 참조의 CreateEnvironmentEc2를 참조하세요.

### `create-environment-membership`

다음 코드 예시에서는 `create-environment-membership`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS Cloud9 개발 환경에 환경 멤버를 추가하려면

이 예제에서는 지정된 Cloud AWS Cloud9 개발 환경에 지정된 환경 멤버를 추가합니다.

명령:

```
aws cloud9 create-environment-membership --environment-
id 8a34f51ce1e04a08882f1e811bd706EX --user-arn arn:aws:iam::123456789012:user/
AnotherDemoUser --permissions read-write
```

출력:

```
{
  "membership": {
```

```
    "environmentId": "8a34f51ce1e04a08882f1e811bd706EX",
    "userId": "AIDAJ3LOROMOUXTBSU6EX",
    "userArn": "arn:aws:iam::123456789012:user/AnotherDemoUser",
    "permissions": "read-write"
  }
}
```

- 자세한 API 내용은 명령 참조CreateEnvironmentMembership의 섹션을 참조하세요. AWS CLI

### `delete-environment-membership`

다음 코드 예시에서는 `delete-environment-membership`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS Cloud9 개발 환경에서 환경 구성원을 삭제하려면

이 예제에서는 지정된 AWS Cloud9 개발 환경에서 지정된 환경 멤버를 삭제합니다.

명령:

```
aws cloud9 delete-environment-membership --environment-
id 8a34f51ce1e04a08882f1e811bd706EX --user-arn arn:aws:iam::123456789012:user/
AnotherDemoUser
```

출력:

```
None.
```

- 자세한 API 내용은 명령 참조DeleteEnvironmentMembership의 섹션을 참조하세요. AWS CLI

### `delete-environment`

다음 코드 예시에서는 `delete-environment`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS Cloud9 개발 환경을 삭제하려면

이 예제에서는 지정된 AWS Cloud9 개발 환경을 삭제합니다. Amazon EC2 인스턴스가 환경에 연결된 경우 도 인스턴스를 종료합니다.

명령:

```
aws cloud9 delete-environment --environment-id 8a34f51ce1e04a08882f1e811bd706EX
```

출력:

```
None.
```

- 자세한 API 내용은 명령 참조DeleteEnvironment의 섹션을 참조하세요. AWS CLI

### describe-environment-memberships

다음 코드 예시에서는 describe-environment-memberships을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS Cloud9 개발 환경의 환경 멤버에 대한 정보를 가져오려면

이 예제에서는 지정된 AWS Cloud9 개발 환경의 환경 멤버에 대한 정보를 가져옵니다.

명령:

```
aws cloud9 describe-environment-memberships --environment-
id 8a34f51ce1e04a08882f1e811bd706EX
```

출력:

```
{
  "memberships": [
    {
      "environmentId": "8a34f51ce1e04a08882f1e811bd706EX",
      "userId": "AIDAJ3LOROMOUXTBSU6EX",
      "userArn": "arn:aws:iam::123456789012:user/AnotherDemoUser",
      "permissions": "read-write"
    },
    {
      "environmentId": "8a34f51ce1e04a08882f1e811bd706EX",
      "userId": "AIDAJNUEDQAQWFELJDLEX",
      "userArn": "arn:aws:iam::123456789012:user/MyDemoUser",
      "permissions": "owner"
    }
  ]
```

```
}
```

AWS Cloud9 개발 환경의 소유자에 대한 정보를 가져오려면

이 예제에서는 지정된 AWS Cloud9 개발 환경의 소유자에 대한 정보를 가져옵니다.

명령:

```
aws cloud9 describe-environment-memberships --environment-
id 8a34f51ce1e04a08882f1e811bd706EX --permissions owner
```

출력:

```
{
  "memberships": [
    {
      "environmentId": "8a34f51ce1e04a08882f1e811bd706EX",
      "userId": "AIDAJNUEDQAQWFELJDLEX",
      "userArn": "arn:aws:iam::123456789012:user/MyDemoUser",
      "permissions": "owner"
    }
  ]
}
```

여러 AWS Cloud9 개발 환경의 환경 멤버에 대한 정보를 가져오려면

이 예제에서는 여러 AWS Cloud9 개발 환경에 지정된 환경 멤버에 대한 정보를 가져옵니다.

명령:

```
aws cloud9 describe-environment-memberships --user-
arn arn:aws:iam::123456789012:user/MyDemoUser
```

출력:

```
{
  "memberships": [
    {
      "environmentId": "10a75714bd494714929e7f5ec4125aEX",
      "lastAccess": 1516213427.0,
      "userId": "AIDAJNUEDQAQWFELJDLEX",
      "userArn": "arn:aws:iam::123456789012:user/MyDemoUser",
```

```
      "permissions": "owner"
    },
    {
      "environmentId": "1980b80e5f584920801c09086667f0EX",
      "lastAccess": 1516144884.0,
      "userId": "AIDAJNUEDQAQWFELJDLEX",
      "userArn": "arn:aws:iam::123456789012:user/MyDemoUser",
      "permissions": "owner"
    }
  ]
}
```

- 자세한 API 내용은 명령 참조DescribeEnvironmentMemberships의 섹션을 참조하세요. AWS CLI

### describe-environment-status

다음 코드 예시에서는 describe-environment-status을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS Cloud9 개발 환경에 대한 상태 정보를 가져오려면

이 예제에서는 지정된 AWS Cloud9 개발 환경에 대한 상태 정보를 가져옵니다.

명령:

```
aws cloud9 describe-environment-status --environment-
id 685f892f431b45c2b28cb69eadcdb0EX
```

출력:

```
{
  "status": "ready",
  "message": "Environment is ready to use"
}
```

- 자세한 API 내용은 명령 참조DescribeEnvironmentStatus의 섹션을 참조하세요. AWS CLI

### describe-environments

다음 코드 예시에서는 describe-environments을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS Cloud9 개발 환경에 대한 정보를 얻으려면

이 예제에서는 지정된 AWS Cloud9 개발 환경에 대한 정보를 가져옵니다.

명령:

```
aws cloud9 describe-environments --environment-
ids 685f892f431b45c2b28cb69eadcdb0EX 349c86d4579e4e7298d500ff57a6b2EX
```

출력:

```
{
  "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"
      }
    }
  ]
}
```

- 자세한 API 내용은 명령 참조DescribeEnvironments의 섹션을 참조하세요. AWS CLI

### `list-environments`

다음 코드 예시에서는 `list-environments`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 AWS Cloud9 개발 환경 식별자 목록을 가져오려면

이 예제에서는 사용 가능한 AWS Cloud9 개발 환경 식별자 목록을 가져옵니다.

명령:

```
aws cloud9 list-environments
```

출력:

```
{
  "environmentIds": [
    "685f892f431b45c2b28cb69eadcdb0EX",
    "1980b80e5f584920801c09086667f0EX"
  ]
}
```

- 자세한 API 내용은 명령 참조ListEnvironments의 섹션을 참조하세요. AWS CLI

### `update-environment-membership`

다음 코드 예시에서는 `update-environment-membership`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS Cloud9 개발 환경에 대한 기존 환경 멤버의 설정을 변경하려면

이 예제에서는 지정된 AWS Cloud9 개발 환경에 대해 지정된 기존 환경 멤버의 설정을 변경합니다.

명령:

```
aws cloud9 update-environment-membership --environment-
id 8a34f51ce1e04a08882f1e811bd706EX --user-arn arn:aws:iam::123456789012:user/
AnotherDemoUser --permissions read-only
```

출력:

```
{
  "membership": {
    "environmentId": "8a34f51ce1e04a08882f1e811bd706EX",
    "userId": "AIDAJ3LOROMOUXTBSU6EX",
    "userArn": "arn:aws:iam::123456789012:user/AnotherDemoUser",
    "permissions": "read-only"
  }
}
```

- 자세한 API 내용은 명령 참조UpdateEnvironmentMembership의 섹션을 참조하세요. AWS CLI

**update-environment**

다음 코드 예시에서는 update-environment을 사용하는 방법을 보여 줍니다.

AWS CLI

기존 AWS Cloud9 개발 환경의 설정을 변경하려면

이 예제에서는 지정된 기존 AWS Cloud9 개발 환경의 지정된 설정을 변경합니다.

명령:

```
aws cloud9 update-environment --environment-id 8a34f51ce1e04a08882f1e811bd706EX
 --name my-changed-demo-env --description "My changed demonstration development
 environment."
```

출력:

```
None.
```

- 자세한 API 내용은 명령 참조UpdateEnvironment의 섹션을 참조하세요. AWS CLI

## AWS CloudFormation 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS CloudFormation.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### `activate-type`

다음 코드 예시에서는 `activate-type`을 사용하는 방법을 보여 줍니다.

AWS CLI

유형을 활성화하려면

다음 `activate-type` 예제에서는 퍼블릭 서드 파티 확장을 활성화하여 스택 템플릿에서 사용할 수 있도록 합니다.

```
aws cloudformation activate-type \
    --region us-west-2 \
    --type RESOURCE \
    --type-name Example::Test::1234567890abcdef0 \
    --type-name-alias Example::Test::Alias
```

출력:

```
{
    "Arn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/Example-
Test-Alias"
}
```

자세한 내용은 AWS CloudFormation 사용 설명서의 AWS CloudFormation 레지스트리 사용을 참조하세요.

- 자세한 API 내용은 명령 참조ActivateType의 섹션을 참조하세요. AWS CLI

### `batch-describe-type-configurations`

다음 코드 예시에서는 `batch-describe-type-configurations`을 사용하는 방법을 보여 줍니다.

AWS CLI

유형 구성을 일괄 설명하는 방법

다음 batch-describe-type-configurations 예제에서는 유형에 대한 데이터를 구성합니다.

```
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"
```

출력:

```
{
    "Errors": [],
    "UnprocessedTypeConfigurations": [],
    "TypeConfigurations": [
        {
            "Arn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/
Example-Test-Type",
            "Alias": "MyConfiguration",
            "Configuration": "{\n        \"Example\": {\n            \"ApiKey\":
 \"examplekey\",\n            \"ApplicationKey\": \"examplekey1\",\n
 \"ApiURL\": \"exampleurl\"\n            }\n}",
            "LastUpdated": "2021-10-01T15:25:46.210000+00:00",
            "TypeArn": "arn:aws:cloudformation:us-east-1:123456789012:type/resource/
Example-Test-Type"
        }
    ]
}
```

자세한 내용은 AWS CloudFormation 사용 설명서의 AWS CloudFormation 레지스트리 사용을 참조하세요.

- 자세한 API 내용은 명령 참조BatchDescribeTypeConfigurations의 섹션을 참조하세요. AWS CLI

## cancel-update-stack

다음 코드 예시에서는 cancel-update-stack을 사용하는 방법을 보여 줍니다.

AWS CLI

진행 중인 스택 업데이트를 취소하려면

다음 cancel-update-stack 명령은 myteststack 스택의 스택 업데이트를 취소합니다.

```
aws cloudformation cancel-update-stack --stack-name myteststack
```

- 자세한 API 내용은 명령 참조CancelUpdateStack의 섹션을 참조하세요. AWS CLI

### continue-update-rollback

다음 코드 예시에서는 continue-update-rollback을 사용하는 방법을 보여 줍니다.

AWS CLI

업데이트 롤백을 재시도하려면

다음 continue-update-rollback 예제는 이전에 실패한 스택 업데이트에서 롤백 작업을 재개합니다.

```
aws cloudformation continue-update-rollback \
    --stack-name my-stack
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조ContinueUpdateRollback의 섹션을 참조하세요. AWS CLI

### create-change-set

다음 코드 예시에서는 create-change-set을 사용하는 방법을 보여 줍니다.

AWS CLI

변경 세트를 생성하려면

다음 create-change-set 예제에서는 CAPABILITY_IAM 기능을 사용하여 변경 세트를 생성합니다. 파일은 현재 폴더의 AWS CloudFormation 템플릿template.yaml으로, IAM 리소스가 포함된 스택을 정의합니다.

```
aws cloudformation create-change-set \
    --stack-name my-application \
```

```
    --change-set-name my-change-set \
    --template-body file://template.yaml \
    --capabilities CAPABILITY_IAM
```

출력:

```
{
    "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"
}
```

- 자세한 API 내용은 명령 참조CreateChangeSet의 섹션을 참조하세요. AWS CLI

### `create-stack-instances`

다음 코드 예시에서는 `create-stack-instances`을 사용하는 방법을 보여 줍니다.

AWS CLI

스택 인스턴스를 생성하려면

다음 `create-stack-instances` 예제에서는 두 계정과 네 리전에 스택 세트의 인스턴스를 생성합니다. 내결함성 설정을 사용하면 일부 스택을 생성할 수 없더라도 모든 계정과 리전에서 업데이트를 시도할 수 있습니다.

```
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
```

출력:

```
{
    "OperationId": "d7995c31-83c2-xmpl-a3d4-e9ca2811563f"
}
```

스택 세트를 생성하려면 `create-stack-set` 명령을 사용합니다.

- 자세한 API 내용은 명령 참조CreateStackInstances의 섹션을 참조하세요. AWS CLI

### create-stack-set

다음 코드 예시에서는 create-stack-set을 사용하는 방법을 보여 줍니다.

AWS CLI

스택 세트를 생성하려면

다음 create-stack-set 예제에서는 지정된 YAML 파일 temlplate를 사용하여 스택 세트를 생성합니다. template.yaml는 스택을 정의하는 현재 폴더의 AWS CloudFormation 템플릿입니다.

```
aws cloudformation create-stack-set \
    --stack-set-name my-stack-set \
    --template-body file://template.yaml \
    --description "SNS topic"
```

출력:

```
{
    "StackSetId": "my-stack-set:8d0f160b-d157-xmpl-a8e6-c0ce8e5d8cc1"
}
```

스택 세트에 스택 인스턴스를 추가하려면 create-stack-instances 명령을 사용합니다.

- 자세한 API 내용은 명령 참조CreateStackSet의 섹션을 참조하세요. AWS CLI

### create-stack

다음 코드 예시에서는 create-stack을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS CloudFormation 스택을 생성하려면

다음 create-stacks 명령에서는 sampletemplate.json 템플릿을 사용하여 이름이 myteststack인 스택을 생성합니다.

```
aws cloudformation create-stack --stack-name myteststack --template-body file://
sampletemplate.json --parameters ParameterKey=KeyPairName,ParameterValue=TestKey
 ParameterKey=SubnetIDs,ParameterValue=SubnetID1\\,SubnetID2
```

출력:

```
{
    "StackId": "arn:aws:cloudformation:us-east-1:123456789012:stack/
myteststack/466df9e0-0dff-08e3-8e2f-5088487c4896"
}
```

자세한 내용은 AWS CloudFormation 사용 설명서의 스택을 참조하세요.

- 자세한 API 내용은 명령 참조CreateStack의 섹션을 참조하세요. AWS CLI

### deactivate-type

다음 코드 예시에서는 deactivate-type을 사용하는 방법을 보여 줍니다.

AWS CLI

유형을 비활성화하려면

다음 deactivate-type 예제에서는 이 계정 및 리전에서 이전에 활성화된 퍼블릭 확장을 비활성화합니다.

```
aws cloudformation deactivate-type \
    --region us-west-2 \
    --type MODULE \
    --type-name Example::Test::Type::MODULE
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS CloudFormation 사용 설명서의 AWS CloudFormation 레지스트리 사용을 참조하세요.

- 자세한 API 내용은 명령 참조DeactivateType의 섹션을 참조하세요. AWS CLI

### delete-change-set

다음 코드 예시에서는 delete-change-set을 사용하는 방법을 보여 줍니다.

AWS CLI

변경 세트를 삭제하려면

다음 delete-change-set 예제에서는 변경 세트 이름과 스택 이름을 지정하여 변경 세트를 삭제합니다.

```
aws cloudformation delete-change-set \
    --stack-name my-stack \
    --change-set-name my-change-set
```

이 명령은 출력을 생성하지 않습니다.

다음 delete-change-set 예제에서는 변경 세트ARN의 전체 를 지정하여 변경 세트를 삭제합니다.

```
aws cloudformation delete-change-set \
    --change-set-name arn:aws:cloudformation:us-east-2:123456789012:changeSet/my-change-set/4eca1a01-e285-xmpl-8026-9a1967bfb4b0
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeleteChangeSet의 섹션을 참조하세요. AWS CLI

### delete-stack-instances

다음 코드 예시에서는 delete-stack-instances을 사용하는 방법을 보여 줍니다.

AWS CLI

스택 인스턴스를 삭제하려면

다음 delete-stack-instances 예제에서는 두 리전의 두 계정에 있는 스택 세트의 인스턴스를 삭제하고 스택을 종료합니다.

```
aws cloudformation delete-stack-instances \
    --stack-set-name my-stack-set \
    --accounts 123456789012 567890123456 \
    --regions us-east-1 us-west-1 \
    --no-retain-stacks
```

출력:

```
{
    "OperationId": "ad49f10c-fd1d-413f-a20a-8de6e2fa8f27"
}
```

빈 스택 세트를 삭제하려면 delete-stack-set 명령을 사용합니다.

- 자세한 API 내용은 명령 참조DeleteStackInstances의 섹션을 참조하세요. AWS CLI

### delete-stack-set

다음 코드 예시에서는 delete-stack-set을 사용하는 방법을 보여 줍니다.

AWS CLI

스택 세트를 삭제하려면

다음 명령은 지정된 빈 스택 세트를 삭제합니다. 스택 세트는 비어 있어야 합니다.

```
aws cloudformation delete-stack-set \
    --stack-set-name my-stack-set
```

이 명령은 출력을 생성하지 않습니다.

스택 세트에서 인스턴스를 삭제하려면 delete-stack-instances 명령을 사용합니다.

- 자세한 API 내용은 명령 참조DeleteStackSet의 섹션을 참조하세요. AWS CLI

### delete-stack

다음 코드 예시에서는 delete-stack을 사용하는 방법을 보여 줍니다.

AWS CLI

스택을 삭제하려면

다음 delete-stack 예제에서는 지정된 스택을 삭제합니다.

```
aws cloudformation delete-stack \
    --stack-name my-stack
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeleteStack의 섹션을 참조하세요. AWS CLI

### deploy

다음 코드 예시에서는 deploy을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 이름이 인 템플릿을 라는 스택template.json에 배포합니다my-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
```

- API 자세한 내용은 AWS CLI 명령 참조의 배포를 참조하세요.

### deregister-type

다음 코드 예시에서는 deregister-type을 사용하는 방법을 보여 줍니다.

AWS CLI

유형 버전 등록을 취소하려면

다음 deregister-type 예제에서는 지정된 유형 버전을 CloudFormation 레지스트리의 활성 사용에서 제거하여 CloudFormation 작업에 더 이상 사용할 수 없도록 합니다.

```
aws cloudformation deregister-type \
    --type RESOURCE \
    --type-name My::Logs::LogGroup \
    --version-id 00000002
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS CloudFormation 사용 설명서의 CloudFormation 레지스트리 사용을 참조하세요.

- 자세한 API 내용은 명령 참조DeregisterType의 섹션을 참조하세요. AWS CLI

### describe-account-limits

다음 코드 예시에서는 describe-account-limits을 사용하는 방법을 보여 줍니다.

AWS CLI

계정 한도에 대한 정보를 가져오려면

다음 명령은 현재 계정에 대한 리전 제한 목록을 검색합니다.

```
aws cloudformation describe-account-limits
```

출력:

```
{
    "AccountLimits": [
        {
            "Name": "StackLimit",
            "Value": 200
        },
        {
            "Name": "StackOutputsLimit",
            "Value": 60
        },
        {
            "Name": "ConcurrentResourcesLimit",
            "Value": 2500
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeAccountLimits의 섹션을 참조하세요. AWS CLI

### describe-change-set

다음 코드 예시에서는 describe-change-set을 사용하는 방법을 보여 줍니다.

AWS CLI

변경 세트에 대한 정보를 가져오려면

다음 describe-change-set 예제에서는 변경 세트 이름 및 스택 이름으로 지정된 변경 세트의 세부 정보를 표시합니다.

```
aws cloudformation describe-change-set \
    --change-set-name my-change-set \
    --stack-name my-stack
```

다음 describe-change-set 예제에서는 전체 변경 세트에 지정된 변경 세트ARN의 세부 정보를 표시합니다.

```
aws cloudformation describe-change-set \
    --change-set-name arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-change-set/bc9555ba-a949-xmpl-bfb8-f41d04ec5784
```

출력:

```
{
    "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
}
```

- 자세한 API 내용은 명령 참조DescribeChangeSet의 섹션을 참조하세요. AWS CLI

### `describe-publisher`

다음 코드 예시에서는 `describe-publisher`을 사용하는 방법을 보여 줍니다.

AWS CLI

게시자를 설명하려면

다음 `describe-publisher` 예제에서는 게시자에 대한 정보를 구성합니다.

```
aws cloudformation describe-publisher \
    --region us-west-2 \
    --publisher-id 000q6TfUovXsEMmgKowxDZLlwqr2QUsh
```

출력:

```
{
    "PublisherId": "000q6TfUovXsEMmgKowxDZLlwqr2QUshd2e75c8c",
    "PublisherStatus": "VERIFIED",
    "IdentityProvider": "AWS_Marketplace",
    "PublisherProfile": "https://aws.amazon.com/marketplace/seller-profile?
id=2c5dc1f0-17cd-4259-8e46-822a83gdtegd"
}
```

자세한 내용은 AWS CloudFormation 사용 설명서의 AWS CloudFormation 레지스트리 사용을 참조하세요.

- 자세한 API 내용은 명령 참조DescribePublisher의 섹션을 참조하세요. AWS CLI

### describe-stack-drift-detection-status

다음 코드 예시에서는 describe-stack-drift-detection-status을 사용하는 방법을 보여 줍니다.

AWS CLI

드리프트 감지 작업의 상태를 확인하려면

다음 describe-stack-drift-detection-status 예제에서는 드리프트 감지 작업의 상태를 표시합니다. detect-stack-drift 명령을 실행하는 ID로 를 가져옵니다.

```
aws cloudformation describe-stack-drift-detection-status \
    --stack-drift-detection-id 1a229160-e4d9-xmpl-ab67-0a4f93df83d4
```

출력:

```
{
    "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"
}
```

- 자세한 API 내용은 명령 참조DescribeStackDriftDetectionStatus의 섹션을 참조하세요. AWS CLI

### describe-stack-events

다음 코드 예시에서는 describe-stack-events을 사용하는 방법을 보여 줍니다.

AWS CLI

스택 이벤트를 설명하려면

다음 describe-stack-events 예제에서는 지정된 스택의 가장 최근 이벤트 2개를 표시합니다.

```
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": "eyJOZXh0VG9XMPLiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ=="
}
```

- 자세한 API 내용은 명령 참조DescribeStackEvents의 섹션을 참조하세요. AWS CLI

### describe-stack-instance

다음 코드 예시에서는 `describe-stack-instance`을 사용하는 방법을 보여 줍니다.

AWS CLI

스택 인스턴스를 설명하려면

다음 명령은 지정된 계정 및 리전에서 지정된 스택 세트의 인스턴스를 설명합니다. 스택 세트는 현재 리전 및 계정에 있고 인스턴스는 계정 의 us-west-2 리전에 있습니다123456789012.:

```
aws cloudformation describe-stack-instance \
    --stack-set-name my-stack-set \
    --stack-instance-account 123456789012 \
    --stack-instance-region us-west-2
```

출력:

```
{
    "StackInstance": {
        "StackSetId": "enable-config:296a3360-xmpl-40af-be78-9341e95bf743",
        "Region": "us-west-2",
        "Account": "123456789012",
        "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/
StackSet-enable-config-e6cac20f-xmpl-46e9-8314-53e0d4591532/4287f9a0-e615-
xmpl-894a-12b31d3117be",
        "ParameterOverrides": [],
        "Status": "OUTDATED",
        "StatusReason": "ResourceLogicalId:ConfigBucket,
 ResourceType:AWS::S3::Bucket, ResourceStatusReason:You have attempted to create
 more buckets than allowed (Service: Amazon S3; Status Code: 400; Error Code:
 TooManyBuckets; Request ID: F7F21CXMPL580224; S3 Extended Request ID: egd/
Fdt89BXMPLyiqbMNljVk55Yqqvi3NYW2nKLUVWhUGEhNfCmZdyj967lhriaG/dWMobSO40o=)."
    }
}
```

- 자세한 API 내용은 명령 참조DescribeStackInstance의 섹션을 참조하세요. AWS CLI

### describe-stack-resource-drifts

다음 코드 예시에서는 describe-stack-resource-drifts을 사용하는 방법을 보여 줍니다.

AWS CLI

스택 정의에서 드리프트된 리소스에 대한 정보를 가져오려면

다음 명령은 지정된 스택의 드리프트된 리소스에 대한 정보를 표시합니다. 드리프트 감지를 시작하려면 detect-stack-drift 명령을 사용합니다.

```
aws cloudformation describe-stack-resource-drifts \
```

```
    --stack-name my-stack
```

출력은 수정된 AWS Lambda 함수를 보여줍니다 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-HIZXMPLEOM9E\",\"Runtime\":
\"nodejs10.x\",\"Tags\":[{\"Key\":\"lambda:createdBy\",\"Value\":\"SAM\"}],\"Timeout
\":900,\"TracingConfig\":{\"Mode\":\"Active\"}}",
            "ActualProperties": "{\"Description\":\"Write a file to S3.\",
\"Environment\":{\"Variables\":{\"bucket\":\"my-stack-bucket-1vc62xmplgguf
\"}},\"Handler\":\"index.handler\",\"MemorySize\":256,\"Role\":
\"arn:aws:iam::123456789012:role/my-functionRole-HIZXMPLEOM9E\",\"Runtime\":
\"nodejs10.x\",\"Tags\":[{\"Key\":\"lambda:createdBy\",\"Value\":\"SAM\"}],\"Timeout
\":22,\"TracingConfig\":{\"Mode\":\"Active\"}}",
            "PropertyDifferences": [
                {
                    "PropertyPath": "/MemorySize",
                    "ExpectedValue": "128",
                    "ActualValue": "256",
                    "DifferenceType": "NOT_EQUAL"
                },
                {
                    "PropertyPath": "/Timeout",
                    "ExpectedValue": "900",
                    "ActualValue": "22",
                    "DifferenceType": "NOT_EQUAL"
                }
            ],
            "StackResourceDriftStatus": "MODIFIED",
            "Timestamp": "2019-10-02T05:54:44.064Z"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeStackResourceDrifts의 섹션을 참조하세요. AWS CLI

### describe-stack-resource

다음 코드 예시에서는 describe-stack-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

스택 리소스에 대한 정보를 가져오려면

다음 describe-stack-resource 예제에서는 지정된 스택의 이름이 MyFunction인 리소스에 대한 세부 정보를 표시합니다.

```
aws cloudformation describe-stack-resource \
    --stack-name MyStack \
    --logical-resource-id MyFunction
```

출력:

```
{
    "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"
        }
    }
}
```

- 자세한 API 내용은 명령 참조DescribeStackResource의 섹션을 참조하세요. AWS CLI

### describe-stack-resources

다음 코드 예시에서는 describe-stack-resources을 사용하는 방법을 보여 줍니다.

AWS CLI

스택 리소스에 대한 정보를 가져오려면

다음 `describe-stack-resources` 예제에서는 지정된 스택의 리소스에 대한 세부 정보를 표시합니다.

```
aws cloudformation describe-stack-resources \
    --stack-name my-stack
```

출력:

```
{
    "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"
            }
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeStackResources의 섹션을 참조하세요. AWS CLI

### `describe-stack-set-operation`

다음 코드 예시에서는 `describe-stack-set-operation`을 사용하는 방법을 보여 줍니다.

AWS CLI

스택 세트 작업에 대한 정보를 가져오려면

다음 describe-stack-set-operation` 예제는 지정된 스택 세트에 대한 업데이트 작업에 대한 세부 정보를 표시합니다.

```
aws cloudformation describe-stack-set-operation \
    --stack-set-name enable-config \
    --operation-id 35d45ebc-ed88-xmpl-ab59-0197a1fc83a0
```

출력:

```
{
    "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"
    }
}
```

- 자세한 API 내용은 명령 참조DescribeStackSetOperation의 섹션을 참조하세요. AWS CLI

### `describe-stack-set`

다음 코드 예시에서는 `describe-stack-set`을 사용하는 방법을 보여 줍니다.

AWS CLI

스택 세트에 대한 정보를 가져오려면

다음 describe-stack-set` 예제는 지정된 스택 세트에 대한 세부 정보를 표시합니다.

```
aws cloudformation describe-stack-set \
    --stack-set-name my-stack-set
```

출력:

```
{
    "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"
    }
}
```

- 자세한 API 내용은 명령 참조DescribeStackSet의 섹션을 참조하세요. AWS CLI

### `describe-stacks`

다음 코드 예시에서는 `describe-stacks`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS CloudFormation 스택을 설명하려면

다음 `describe-stacks` 명령에서는 `myteststack` 스택에 대한 요약 정보를 보여줍니다.

```
aws cloudformation describe-stacks --stack-name myteststack
```

출력:

```
{
    "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-jssofi1zie2w"
                }
            ],
```

```
            "StackStatusReason": null,
            "CreationTime": "2013-08-23T01:02:15.422Z",
            "Capabilities": [],
            "StackName": "myteststack",
            "StackStatus": "CREATE_COMPLETE",
            "DisableRollback": false
        }
    ]
}
```

자세한 내용은 AWS CloudFormation 사용 설명서의 스택을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeStacks의 섹션을 참조하세요. AWS CLI

### `describe-type-registration`

다음 코드 예시에서는 `describe-type-registration`을 사용하는 방법을 보여 줍니다.

AWS CLI

유형 등록 정보를 표시하려면

다음 `describe-type-registration` 예제에서는 유형의 현재 상태, 유형 및 버전을 포함하여 지정된 유형 등록에 대한 정보를 표시합니다.

```
aws cloudformation describe-type-registration \
    --registration-token a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS CloudFormation 사용 설명서의 CloudFormation 레지스트리 사용을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeTypeRegistration의 섹션을 참조하세요. AWS CLI

### describe-type

다음 코드 예시에서는 describe-type을 사용하는 방법을 보여 줍니다.

AWS CLI

유형 정보를 표시하려면

다음 describe-type 예제에서는 지정된 유형에 대한 정보를 표시합니다.

```
aws cloudformation describe-type \
    --type-name My::Logs::LogGroup \
    --type RESOURCE
```

출력:

```
{
    "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]"
}
```

자세한 내용은 AWS CloudFormation 사용 설명서의 CloudFormation 레지스트리 사용을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeType의 섹션을 참조하세요. AWS CLI

### detect-stack-drift

다음 코드 예시에서는 detect-stack-drift을 사용하는 방법을 보여 줍니다.

AWS CLI

드리프트된 리소스를 감지하려면

다음 `detect-stack-drift` 예제에서는 지정된 스택에 대한 드리프트 감지를 시작합니다.

```
aws cloudformation detect-stack-drift \
    --stack-name my-stack
```

출력:

```
{
    "StackDriftDetectionId": "1a229160-e4d9-xmpl-ab67-0a4f93df83d4"
}
```

그런 다음 이 ID를 `describe-stack-resource-drifts` 명령과 함께 사용하여 드리프트된 리소스를 설명할 수 있습니다.

- 자세한 API 내용은 명령 참조DetectStackDrift의 섹션을 참조하세요. AWS CLI

### `detect-stack-resource-drift`

다음 코드 예시에서는 `detect-stack-resource-drift`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스의 드리프트를 감지하려면

다음 `detect-stack-resource-drift` 예제에서는 드리프트에 `MyStack` 대해 이라는 스택`MyFunction`에 이름이 지정된 리소스를 확인합니다.

```
aws cloudformation detect-stack-resource-drift \
   --stack-name MyStack \
   --logical-resource-id MyFunction
```

출력은 수정된 AWS Lambda 함수를 보여줍니다 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-HIZXMPLEOM9E\",\"Runtime\":
\"nodejs10.x\",\"Tags\":[{\"Key\":\"lambda:createdBy\",\"Value\":\"SAM\"}],\"Timeout
\":900,\"TracingConfig\":{\"Mode\":\"Active\"}}",
        "ActualProperties": "{\"Description\":\"Write a file to S3.\",\"Environment
\":{\"Variables\":{\"bucket\":\"my-stack-bucket-1vc62xmplgguf\"}},\"Handler\":
\"index.handler\",\"MemorySize\":256,\"Role\":\"arn:aws:iam::123456789012:role/
my-functionRole-HIZXMPLEOM9E\",\"Runtime\":\"nodejs10.x\",\"Tags\":[{\"Key\":
\"lambda:createdBy\",\"Value\":\"SAM\"}],\"Timeout\":22,\"TracingConfig\":{\"Mode\":
\"Active\"}}",
        "PropertyDifferences": [
            {
                "PropertyPath": "/MemorySize",
                "ExpectedValue": "128",
                "ActualValue": "256",
                "DifferenceType": "NOT_EQUAL"
            },
            {
                "PropertyPath": "/Timeout",
                "ExpectedValue": "900",
                "ActualValue": "22",
                "DifferenceType": "NOT_EQUAL"
            }
        ],
        "StackResourceDriftStatus": "MODIFIED",
        "Timestamp": "2019-10-02T05:58:47.433Z"
    }
}
```

- 자세한 API 내용은 명령 참조DetectStackResourceDrift의 섹션을 참조하세요. AWS CLI

### `detect-stack-set-drift`

다음 코드 예시에서는 `detect-stack-set-drift`을 사용하는 방법을 보여 줍니다.

AWS CLI

스택 세트 및 연결된 모든 스택 인스턴스에서 드리프트를 감지하려면

다음 detect-stack-set-drift 예제에서는 해당 스택 세트와 연결된 모든 스택 인스턴스를 포함하여 지정된 스택 세트에서 드리프트 감지 작업을 시작하고 드리프트 작업의 상태를 추적하는 데 사용할 수 있는 작업 ID를 반환합니다.

```
aws cloudformation detect-stack-set-drift \
    --stack-set-name stack-set-drift-example
```

출력:

```
{
    "OperationId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

자세한 내용은 AWS CloudFormation 사용 설명서의 스택 세트에서 관리되지 않는 구성 변경 사항 감지를 참조하세요.

- 자세한 API 내용은 명령 참조DetectStackSetDrift의 섹션을 참조하세요. AWS CLI

### estimate-template-cost

다음 코드 예시에서는 estimate-template-cost을 사용하는 방법을 보여 줍니다.

AWS CLI

템플릿 비용을 추정하려면

다음 estimate-template-cost 예제에서는 현재 폴더에서 이름이 template.yaml인 템플릿에 대한 예상 비용을 생성합니다.

```
aws cloudformation estimate-template-cost \
    --template-body file://template.yaml
```

출력:

```
{
    "Url": "http://calculator.s3.amazonaws.com/calc5.html?
key=cloudformation/7870825a-xmpl-4def-92e7-c4f8dd360cca"
}
```

- 자세한 API 내용은 명령 참조EstimateTemplateCost의 섹션을 참조하세요. AWS CLI

### execute-change-set

다음 코드 예시에서는 execute-change-set을 사용하는 방법을 보여 줍니다.

AWS CLI

변경 세트를 실행하려면

다음 execute-change-set 예제에서는 변경 세트 이름 및 스택 이름으로 지정된 변경 세트를 실행합니다.

```
aws cloudformation execute-change-set \
    --change-set-name my-change-set \
    --stack-name my-stack
```

다음 execute-change-set 예제에서는 전체 변경 세트에서 지정한 ARN 변경 세트를 실행합니다.

```
aws cloudformation execute-change-set \
    --change-set-name arn:aws:cloudformation:us-west-2:123456789012:changeSet/my-change-set/bc9555ba-a949-xmpl-bfb8-f41d04ec5784
```

- 자세한 API 내용은 명령 참조ExecuteChangeSet의 섹션을 참조하세요. AWS CLI

### get-stack-policy

다음 코드 예시에서는 get-stack-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

스택 정책을 보려면

다음 get-stack-policy 예제에서는 지정된 스택에 대한 스택 정책을 표시합니다. 스택에 정책을 연결하려면 set-stack-policy 명령을 사용합니다.

```
aws cloudformation get-stack-policy \
    --stack-name my-stack
```

출력:

```
{
```

```
    "StackPolicyBody": "{\n  \"Statement\" : [\n    {\n      \"Effect\" :
 \"Allow\",\n      \"Action\" : \"Update:*\",\n      \"Principal\": \"*\",\n
     \"Resource\" : \"*\"\n    },\n    {\n      \"Effect\" : \"Deny\",\n
 \"Action\" : \"Update:*\",\n      \"Principal\": \"*\",\n      \"Resource\" :
 \"LogicalResourceId/bucket\"\n    }\n  ]\n}\n"
}
```

- 자세한 API 내용은 명령 참조GetStackPolicy의 섹션을 참조하세요. AWS CLI

### get-template-summary

다음 코드 예시에서는 get-template-summary을 사용하는 방법을 보여 줍니다.

AWS CLI

템플릿 요약을 표시하려면

다음 명령은 지정된 템플릿 파일의 리소스 및 메타데이터에 대한 요약 정보를 표시합니다.

```
aws cloudformation get-template-summary \
   --template-body file://template.yaml
```

출력:

```
{
    "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"
}
```

- 자세한 API 내용은 명령 참조GetTemplateSummary의 섹션을 참조하세요. AWS CLI

### get-template

다음 코드 예시에서는 get-template을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS CloudFormation 스택의 템플릿 본문을 보려면

다음 get-template 명령에서는 myteststack 스택에 대한 템플릿을 보여줍니다.

```
aws cloudformation get-template --stack-name myteststack
```

출력:

```
{
    "TemplateBody": {
        "AWSTemplateFormatVersion": "2010-09-09",
        "Outputs": {
            "BucketName": {
                "Description": "Name of S3 bucket to hold website content",
                "Value": {
                    "Ref": "S3Bucket"
                }
            }
        },
        "Description": "AWS CloudFormation Sample Template S3_Bucket: Sample
 template showing how to create a publicly accessible S3 bucket. **WARNING** This
 template creates an S3 bucket. You will be billed for the AWS resources used if you
 create a stack from this template.",
        "Resources": {
            "S3Bucket": {
                "Type": "AWS::S3::Bucket",
                "Properties": {
                    "AccessControl": "PublicRead"
                }
            }
        }
    }
}
```

- 자세한 API 내용은 명령 참조GetTemplate의 섹션을 참조하세요. AWS CLI

### list-change-sets

다음 코드 예시에서는 list-change-sets을 사용하는 방법을 보여 줍니다.

AWS CLI

변경 세트를 나열하려면

다음 list-change-sets 예제에서는 지정된 스택에 대해 보류 중인 변경 세트 목록을 표시합니다.

```
aws cloudformation list-change-sets \
   --stack-name my-stack
```

출력:

```
{
    "Summaries": [
        {
            "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-
stack/d0a825a0-e4cd-xmpl-b9fb-061c69e99204",
            "StackName": "my-stack",
            "ChangeSetId": "arn:aws:cloudformation:us-west-2:123456789012:changeSet/
my-change-set/70160340-7914-xmpl-bcbf-128a1fa78b5d",
            "ChangeSetName": "my-change-set",
            "ExecutionStatus": "AVAILABLE",
            "Status": "CREATE_COMPLETE",
            "CreationTime": "2019-10-02T05:38:54.297Z"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListChangeSets의 섹션을 참조하세요. AWS CLI

### list-exports

다음 코드 예시에서는 list-exports을 사용하는 방법을 보여 줍니다.

AWS CLI

내보내기를 나열하려면

다음 list-exports 예제에서는 현재 리전의 스택에서 내보내기 목록을 표시합니다.

```
aws cloudformation list-exports
```

출력:

```
{
    "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-075ed3xmplebd2fb1"
        },
        {
            "ExportingStackId": "arn:aws:cloudformation:us-
west-2:123456789012:stack/private-vpc/99764070-b56c-xmpl-bee8-062a88d1d800",
            "Name": "private-vpc-vpcid",
            "Value": "vpc-011d7xmpl100e9841"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListExports의 섹션을 참조하세요. AWS CLI

### `list-imports`

다음 코드 예시에서는 `list-imports`을 사용하는 방법을 보여 줍니다.

AWS CLI

가져오기를 나열하려면

다음 `list-imports` 예제에서는 지정된 내보내기를 가져오는 스택을 나열합니다. 사용 가능한 내보내기 목록을 가져오려면 `list-exports` 명령을 사용합니다.

```
aws cloudformation list-imports \
```

```
    --export-name private-vpc-vpcid
```

출력:

```
{
    "Imports": [
        "my-database-stack"
    ]
}
```

- 자세한 API 내용은 명령 참조ListImports의 섹션을 참조하세요. AWS CLI

### list-stack-instances

다음 코드 예시에서는 `list-stack-instances`을 사용하는 방법을 보여 줍니다.

AWS CLI

스택의 인스턴스를 나열하려면

다음 `list-stack-instances` 예제에서는 지정된 스택 세트에서 생성된 인스턴스를 나열합니다.

```
aws cloudformation list-stack-instances \
    --stack-set-name enable-config
```

예제 출력에는 오류로 인해 업데이트하지 못한 스택에 대한 세부 정보가 포함됩니다.

```
{
    "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-1OJWJ7XD59WR0' 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)."
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListStackInstances의 섹션을 참조하세요. AWS CLI

### `list-stack-resources`

다음 코드 예시에서는 `list-stack-resources`을 사용하는 방법을 보여 줍니다.

AWS CLI

스택의 리소스를 나열하려면

다음 명령은 지정된 스택의 리소스 목록을 표시합니다.

```
aws cloudformation list-stack-resources \
    --stack-name my-stack
```

출력:

```
{
    "StackResourceSummaries": [
        {
            "LogicalResourceId": "bucket",
            "PhysicalResourceId": "my-stack-bucket-1vc62xmplgguf",
            "ResourceType": "AWS::S3::Bucket",
            "LastUpdatedTimestamp": "2019-10-02T04:34:11.345Z",
            "ResourceStatus": "CREATE_COMPLETE",
            "DriftInformation": {
                "StackResourceDriftStatus": "IN_SYNC"
```

```
            }
        },
        {
            "LogicalResourceId": "function",
            "PhysicalResourceId": "my-function-SEZV4XMPL4S5",
            "ResourceType": "AWS::Lambda::Function",
            "LastUpdatedTimestamp": "2019-10-02T05:34:27.989Z",
            "ResourceStatus": "UPDATE_COMPLETE",
            "DriftInformation": {
                "StackResourceDriftStatus": "IN_SYNC"
            }
        },
        {
            "LogicalResourceId": "functionRole",
            "PhysicalResourceId": "my-functionRole-HIZXMPLEOM9E",
            "ResourceType": "AWS::IAM::Role",
            "LastUpdatedTimestamp": "2019-10-02T04:34:06.350Z",
            "ResourceStatus": "CREATE_COMPLETE",
            "DriftInformation": {
                "StackResourceDriftStatus": "IN_SYNC"
            }
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListStackResources의 섹션을 참조하세요. AWS CLI

### `list-stack-set-operation-results`

다음 코드 예시에서는 `list-stack-set-operation-results`을 사용하는 방법을 보여 줍니다.

AWS CLI

스택 세트 작업 결과를 나열하려면

다음 명령은 지정된 스택 세트의 인스턴스에 대한 업데이트 작업 결과를 표시합니다.

```
aws cloudformation list-stack-set-operation-results \
    --stack-set-name enable-config \
    --operation-id 35d45ebc-ed88-xmpl-ab59-0197a1fc83a0
```

출력:

```
{
    "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"
        }
    ]
}
```

참고: 계정 게이트 함수를 생성하지 않는 한 의 SKIPPED 상태는 성공적인 작업에 대해 AccountGateResult 예상됩니다.

- 자세한 API 내용은 명령 참조ListStackSetOperationResults의 섹션을 참조하세요. AWS CLI

### list-stack-set-operations

다음 코드 예시에서는 list-stack-set-operations을 사용하는 방법을 보여 줍니다.

AWS CLI

스택 세트 작업을 나열하려면

다음 list-stack-set-operations 예제에서는 지정된 스택 세트에 대한 최신 작업 목록을 표시합니다.

```
aws cloudformation list-stack-set-operations \
    --stack-set-name my-stack-set
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListStackSetOperations의 섹션을 참조하세요. AWS CLI

### `list-stack-sets`

다음 코드 예시에서는 `list-stack-sets`을 사용하는 방법을 보여 줍니다.

AWS CLI

스택 세트를 나열하려면

다음 `list-stack-sets` 예제에서는 현재 리전 및 계정의 스택 세트 목록을 표시합니다.

```
aws cloudformation list-stack-sets
```

출력:

```
{
    "Summaries": [
        {
            "StackSetName": "enable-config",
            "StackSetId": "enable-config:296a3360-xmpl-40af-be78-9341e95bf743",
            "Description": "Enable AWS Config",
            "Status": "ACTIVE"
```

```
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListStackSets의 섹션을 참조하세요. AWS CLI

### `list-stacks`

다음 코드 예시에서는 `list-stacks`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS CloudFormation 스택을 나열하려면

다음 `list-stacks` 명령에서는 상태가 `CREATE_COMPLETE`인 모든 스택에 대한 요약 내용을 보여 줍니다.

```
aws cloudformation list-stacks --stack-status-filter CREATE_COMPLETE
```

출력:

```
[
    {
        "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"
    }
]
```

- 자세한 API 내용은 명령 참조ListStacks의 섹션을 참조하세요. AWS CLI

### `list-type-registrations`

다음 코드 예시에서는 `list-type-registrations`을 사용하는 방법을 보여 줍니다.

AWS CLI

유형의 완료된 등록을 나열하려면

다음 `list-type-registrations` 예제에서는 지정된 유형에 대해 완료된 유형 등록 목록을 표시합니다.

```
aws cloudformation list-type-registrations \
    --type RESOURCE \
    --type-name My::Logs::LogGroup \
    --registration-status-filter COMPLETE
```

출력:

```
{
    "RegistrationTokenList": [
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
        "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333"
    ]
}
```

자세한 내용은 AWS CloudFormation 사용 설명서의 CloudFormation 레지스트리 사용을 참조하세요.

- 자세한 API 내용은 명령 참조ListTypeRegistrations의 섹션을 참조하세요. AWS CLI

### `list-type-versions`

다음 코드 예시에서는 `list-type-versions`을 사용하는 방법을 보여 줍니다.

AWS CLI

익스텐션 버전을 나열하려면

다음 `list-type-versions` 예제에서는 확장 버전에 대한 요약 정보를 반환합니다.

```
aws cloudformation list-type-versions \
  --endpoint https://example.com \
  --region us-west-2 \
  --type RESOURCE \
  --type-name My::Resource::Example \
```

```
  --publisher-id 123456789012
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS CloudFormation 사용 설명서의 AWS CloudFormation 레지스트리 사용을 참조하세요.

- 자세한 API 내용은 명령 참조ListTypeVersions의 섹션을 참조하세요. AWS CLI

### `list-types`

다음 코드 예시에서는 `list-types`을 사용하는 방법을 보여 줍니다.

AWS CLI

계정의 프라이빗 리소스 유형을 나열하려면

다음 `list-types` 예제에서는 현재 AWS 계정에 현재 등록된 프라이빗 리소스 유형의 목록을 표시합니다.

```
aws cloudformation list-types
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS CloudFormation 사용 설명서의 CloudFormation 레지스트리 사용을 참조하세요.

- 자세한 API 내용은 명령 참조ListTypes의 섹션을 참조하세요. AWS CLI

### package

다음 코드 예시에서는 package을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 로컬 아티팩트를 S3 버킷에 업로드template.json하여 라는 템플릿을 내보내bucket-name고 내보낸 템플릿을 에 씁니다packaged-template.json.

```
aws cloudformation package --template-file /path_to_template/template.json --s3-
bucket bucket-name --output-template-file packaged-template.json --use-json
```

- API 자세한 내용은 명령 참조의 패키지를 참조하세요. AWS CLI

### publish-type

다음 코드 예시에서는 publish-type을 사용하는 방법을 보여 줍니다.

AWS CLI

확장을 게시하려면

다음 publish-type 예제에서는 지정된 확장을 이 리전의 퍼블릭 확장으로 CloudFormation 레지스트리에 게시합니다.

```
aws cloudformation publish-type \
  --region us-west-2 \
  --type RESOURCE \
  --type-name Example::Test::1234567890abcdef0
```

출력:

```
{
    "PublicTypeArn":"arn:aws:cloudformation:us-west-2::type/
resource/000q6TfUovXsEMmgKowxDZLlwqr2QUshd2e75c8c/Example-
Test-1234567890abcdef0/1.0.0"
}
```

자세한 내용은 AWS CloudFormation 사용 설명서의 AWS CloudFormation 레지스트리 사용을 참조하세요.

- 자세한 API 내용은 명령 참조PublishType의 섹션을 참조하세요. AWS CLI

### register-publisher

다음 코드 예시에서는 register-publisher을 사용하는 방법을 보여 줍니다.

AWS CLI

게시자를 등록하려면

다음 register-publisher 예제에서는 게시자를 등록하고 용어 및 조건 파라미터를 수락합니다.

```
aws cloudformation register-publisher \
  --region us-west-2 \
  --accept-terms-and-conditions
```

출력:

```
{
    "PublisherId": "000q6TfUovXsEMmgKowxDZLlwqr2QUshd2e75c8c"
}
```

자세한 내용은 AWS CloudFormation 사용 설명서의 AWS CloudFormation 레지스트리 사용을 참조하세요.

- 자세한 API 내용은 명령 참조RegisterPublisher의 섹션을 참조하세요. AWS CLI

### register-type

다음 코드 예시에서는 register-type을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 유형을 등록하려면

다음 `register-type` 예제에서는 지정된 리소스 유형을 사용자 계정의 프라이빗 리소스 유형으로 등록합니다.

```
aws cloudformation register-type \
    --type-name My::Organization::ResourceName \
    --schema-handler-package s3://bucket_name/my-organization-resource_name.zip \
    --type RESOURCE
```

출력:

```
{
    "RegistrationToken": "f5525280-104e-4d35-bef5-8f1f1example"
}
```

자세한 내용은 CloudFormation 유형 개발을 위한 명령줄 인터페이스 사용 설명서의 리소스 공급자 등록을 참조하세요.

- 자세한 API 내용은 명령 참조RegisterType의 섹션을 참조하세요. AWS CLI

### `set-stack-policy`

다음 코드 예시에서는 `set-stack-policy`을 사용하는 방법을 보여 줍니다.

AWS CLI

스택 정책을 적용하려면

다음 `set-stack-policy` 예제에서는 지정된 스택의 지정된 리소스에 대한 업데이트를 비활성화합니다. `stack-policy.json`는 스택의 리소스에 허용되는 작업을 정의하는 JSON 문서입니다.

```
aws cloudformation set-stack-policy \
    --stack-name my-stack \
    --stack-policy-body file://stack-policy.json
```

출력:

```
{
```

```
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "*"
    },
    {
      "Effect" : "Deny",
      "Action" : "Update:*",
      "Principal": "*",
      "Resource" : "LogicalResourceId/bucket"
    }
  ]
}
```

- 자세한 API 내용은 명령 참조SetStackPolicy의 섹션을 참조하세요. AWS CLI

### set-type-configuration

다음 코드 예시에서는 set-type-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터를 구성하려면

다음 set-type-configuration 예제에서는 지정된 계정 및 리전의 등록된 CloudFormation 확장에 대한 구성 데이터를 지정합니다.

```
aws cloudformation set-type-configuration \
    --region us-west-2 \
    --type RESOURCE \
    --type-name Example::Test::Type \
    --configuration-alias default \
    --configuration "{\"CredentialKey\": \"testUserCredential\"}"
```

출력:

```
{
    "ConfigurationArn": "arn:aws:cloudformation:us-west-2:123456789012:type-
configuration/resource/Example-Test-Type/default"
```

```
}
```

자세한 내용은 AWS CloudFormation 사용 설명서의 AWS CloudFormation 레지스트리 사용을 참조하세요.

- 자세한 API 내용은 명령 참조SetTypeConfiguration의 섹션을 참조하세요. AWS CLI

### `set-type-default-version`

다음 코드 예시에서는 `set-type-default-version`을 사용하는 방법을 보여 줍니다.

AWS CLI

유형의 기본 버전을 설정하려면

다음 `set-type-default-version` 예제에서는 지정된 유형 버전을 이 유형의 기본값으로 설정합니다.

```
aws cloudformation set-type-default-version \
    --type RESOURCE \
    --type-name My::Logs::LogGroup \
    --version-id 00000003
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS CloudFormation 사용 설명서의 CloudFormation 레지스트리 사용을 참조하세요.

- 자세한 API 내용은 명령 참조SetTypeDefaultVersion의 섹션을 참조하세요. AWS CLI

### `signal-resource`

다음 코드 예시에서는 `signal-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 신호를 보내려면

다음 `signal-resource` 예제는 라는 스택`MyWaitCondition`에 라는 대기 조건을 충족`success`하도록 신호를 보냅니다`my-stack`.

```
aws cloudformation signal-resource \
    --stack-name my-stack \
    --logical-resource-id MyWaitCondition \
    --unique-id 1234 \
    --status SUCCESS
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조SignalResource의 섹션을 참조하세요. AWS CLI

### stop-stack-set-operation

다음 코드 예시에서는 stop-stack-set-operation을 사용하는 방법을 보여 줍니다.

AWS CLI

스택 세트 작업을 중지하려면

다음 stop-stack-set-operation 예제에서는 지정된 스택 세트에 대한 프로모션 내 업데이트 작업을 중지합니다.

```
aws cloudformation stop-stack-set-operation \
    --stack-set-name my-stack-set \
    --operation-id 1261cd27-490b-xmpl-ab42-793a896c69e6
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조StopStackSetOperation의 섹션을 참조하세요. AWS CLI

### test-type

다음 코드 예시에서는 test-type을 사용하는 방법을 보여 줍니다.

AWS CLI

확장을 테스트하려면

다음 test-type 예제에서는 등록된 확장을 테스트하여 CloudFormation 레지스트리에 게시하는 데 필요한 모든 요구 사항을 충족하는지 확인합니다.

```
aws cloudformation test-type \
```

```
    --arn arn:aws:cloudformation:us-west-2:123456789012:type/resource/Sample-Test-
Resource123/00000001
```

출력:

```
{
    "TypeVersionArn": "arn:aws:cloudformation:us-west-2:123456789012:type/resource/
Sample-Test-Resource123/00000001"
}
```

자세한 내용은 AWS CloudFormation 사용 설명서의 AWS CloudFormation 레지스트리 사용을 참조하세요.

- 자세한 API 내용은 명령 참조TestType의 섹션을 참조하세요. AWS CLI

### update-stack-instances

다음 코드 예시에서는 update-stack-instances을 사용하는 방법을 보여 줍니다.

AWS CLI

스택 인스턴스를 업데이트하려면

다음 update-stack-instances 예제에서는 최신 설정을 사용하여 두 리전의 두 계정에 있는 스택 인스턴스에 대한 업데이트를 재시도합니다. 지정된 내결함성 설정을 사용하면 일부 스택을 업데이트할 수 없더라도 모든 계정과 리전에서 업데이트를 시도할 수 있습니다.

```
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
```

출력:

```
{
    "OperationId": "103ebdf2-21ea-xmpl-8892-de5e30733132"
}
```

- 자세한 API 내용은 명령 참조UpdateStackInstances의 섹션을 참조하세요. AWS CLI

### update-stack-set

다음 코드 예시에서는 update-stack-set을 사용하는 방법을 보여 줍니다.

AWS CLI

스택 세트를 업데이트하려면

다음 update-stack-set 예제에서는 키 이름과 값이 Owner 인 태그를 지정된 스택 세트IT의 스택 인스턴스에 추가합니다.

```
aws cloudformation update-stack-set \
    --stack-set-name my-stack-set \
    --use-previous-template \
    --tags Key=Owner,Value=IT
```

출력:

```
{
    "OperationId": "e2b60321-6cab-xmpl-bde7-530c6f47950e"
}
```

- 자세한 API 내용은 명령 참조UpdateStackSet의 섹션을 참조하세요. AWS CLI

### update-stack

다음 코드 예시에서는 update-stack을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS CloudFormation 스택을 업데이트하려면

다음 update-stack 명령에서는 mystack 스택의 템플릿 및 입력 파라미터를 업데이트합니다.

```
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
```

다음 update-stack 명령에서는 mystack 스택의 SubnetIDs 파라미터값만 업데이트합니다. 파라미터값을 지정하지 않으면 템플릿에 지정된 기본값이 사용됩니다.

```
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
```

다음 update-stack 명령에서는 mystack 스택에 스택 알림 주제 2개를 추가합니다.

```
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"
```

자세한 내용은 AWS CloudFormation 사용 설명서의 AWS CloudFormation 스택 업데이트를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateStack의 섹션을 참조하세요. AWS CLI

### update-termination-protection

다음 코드 예시에서는 update-termination-protection을 사용하는 방법을 보여 줍니다.

AWS CLI

종료 방지를 활성화하려면

다음 update-termination-protection 예제에서는 지정된 스택에서 종료 방지를 활성화합니다.

```
aws cloudformation update-termination-protection \
    --stack-name my-stack \
    --enable-termination-protection
```

출력:

```
{
    "StackId": "arn:aws:cloudformation:us-west-2:123456789012:stack/my-stack/
d0a825a0-e4cd-xmpl-b9fb-061c69e99204"
}
```

- 자세한 API 내용은 명령 참조UpdateTerminationProtection의 섹션을 참조하세요. AWS CLI

**validate-template**

다음 코드 예시에서는 validate-template을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS CloudFormation 템플릿을 검증하려면

다음 validate-template 명령은 sampletemplate.json 템플릿의 유효성을 확인합니다.

```
aws cloudformation validate-template --template-body file://sampletemplate.json
```

출력:

```
{
    "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": []
}
```

자세한 내용은 AWS CloudFormation 사용 설명서의 AWS CloudFormation 템플릿 작업을 참조하세요.

- 자세한 API 내용은 명령 참조ValidateTemplate의 섹션을 참조하세요. AWS CLI

## CloudFront 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 CloudFront.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### `create-cloud-front-origin-access-identity`

다음 코드 예시에서는 `create-cloud-front-origin-access-identity`을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudFront 오리진 액세스 자격 증명을 생성하려면

다음 예제에서는 OAI 구성을 명령줄 인수로 제공하여 CloudFront 오리진 액세스 자격 증명(OAI)을 생성합니다.

```
aws cloudfront create-cloud-front-origin-access-identity \
    --cloud-front-origin-access-identity-config \
        CallerReference="cli-example",Comment="Example OAI"
```

다음 예제와 같이 JSON 파일에 OAI 구성을 제공하여 동일한 작업을 수행할 수 있습니다.

```
aws cloudfront create-cloud-front-origin-access-identity \
    --cloud-front-origin-access-identity-config file://OAI-config.json
```

파일은 현재 디렉터리의 JSON 문서`OAI-config.json`로, 다음을 포함합니다.

```
{
    "CallerReference": "cli-example",
    "Comment": "Example OAI"
}
```

OAI 구성에 명령줄 인수를 제공하든 JSON 파일을 제공하든 출력은 동일합니다.

```
{
    "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"
        }
    }
}
```

- 자세한 API 내용은 명령 참조CreateCloudFrontOriginAccessIdentity의 섹션을 참조하세요. AWS CLI

### `create-distribution-with-tags`

다음 코드 예시에서는 `create-distribution-with-tags`을 사용하는 방법을 보여 줍니다.

AWS CLI

태그를 사용하여 CloudFront 배포를 생성하려면

다음 예제에서는 라는 JSON 파일에 배포 구성과 태그를 제공하여 두 개의 태그가 있는 배포를 생성합니다`dist-config-with-tags.json`.

```
aws cloudfront create-distribution-with-tags \
    --distribution-config-with-tags file://dist-config-with-tags.json
```

파일은 현재 폴더의 JSON 문서`dist-config-with-tags.json`로 다음을 포함합니다. 두 개의 태그가 포함된 파일 상단의 `Tags` 객체를 기록해 둡니다.

`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
    }
}
```

출력:

```
{
    "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": "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
        }
    }
}
```

- 자세한 API 내용은 명령 참조CreateDistributionWithTags의 섹션을 참조하세요. AWS CLI

### `create-distribution`

다음 코드 예시에서는 `create-distribution`을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudFront 배포를 생성하려면

다음 예시에서는 다음 명령줄 인수를 사용하여 이름이 `awsexamplebucket`인 S3 버킷에 대한 배포를 생성하고 `index.html`을 기본 루트 객체로 지정합니다.

```
aws cloudfront create-distribution \
    --origin-domain-name awsexamplebucket.s3.amazonaws.com \
    --default-root-object index.html
```

다음 예제와 같이 명령줄 인수를 사용하는 대신 JSON 파일에 배포 구성을 제공할 수 있습니다.

```
aws cloudfront create-distribution \
    --distribution-config file://dist-config.json
```

파일은 현재 폴더의 JSON 문서`dist-config.json`로 다음을 포함합니다.

```
{
    "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
}
```

배포 정보에 명령줄 인수를 제공하든 JSON 파일을 제공하든 출력은 동일합니다.

```
{
    "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
        }
    }
}
```

- 자세한 API 내용은 명령 참조CreateDistribution의 섹션을 참조하세요. AWS CLI

### `create-field-level-encryption-config`

다음 코드 예시에서는 `create-field-level-encryption-config`을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudFront 필드 수준 암호화 구성을 생성하려면

다음 예제에서는 라는 JSON 파일에 구성 파라미터를 제공하여 필드 수준 암호화 구성을 생성합니다`fle-config.json`. 필드 수준 암호화 구성을 생성하려면 먼저 필드 수준 암호화 프로파일이 있어야 합니다. 프로필을 생성하려면 create-field-level-encryption-profile 명령을 참조하세요.

CloudFront 필드 수준 암호화에 대한 자세한 내용은 Amazon CloudFront 개발자 안내서의 필드 수준 암호화를 사용하여 민감한 데이터 보호를 참조하세요.

```
aws cloudfront create-field-level-encryption-config \
    --field-level-encryption-config file://fle-config.json
```

파일은 현재 폴더의 JSON 문서fle-config.json로, 다음을 포함합니다.

```
{
    "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"
                }
            ]
        }
    }
}
```

출력:

```
{
    "Location": "https://cloudfront.amazonaws.com/2019-03-26/field-level-encryption/
C3KM2WVD605UAY",
    "ETag": "E2P4Z4VU7TY5SG",
    "FieldLevelEncryption": {
        "Id": "C3KM2WVD605UAY",
        "LastModifiedTime": "2019-12-10T21:30:18.974Z",
        "FieldLevelEncryptionConfig": {
            "CallerReference": "cli-example",
            "Comment": "Example FLE configuration",
            "QueryArgProfileConfig": {
                "ForwardWhenQueryArgProfileIsUnknown": true,
                "QueryArgProfiles": {
                    "Quantity": 0,
                    "Items": []
```

```
                }
            },
            "ContentTypeProfileConfig": {
                "ForwardWhenContentTypeIsUnknown": true,
                "ContentTypeProfiles": {
                    "Quantity": 1,
                    "Items": [
                        {
                            "Format": "URLEncoded",
                            "ProfileId": "P280MFCLSYOCVU",
                            "ContentType": "application/x-www-form-urlencoded"
                        }
                    ]
                }
            }
        }
    }
}
```

- 자세한 API 내용은 명령 참조CreateFieldLevelEncryptionConfig의 섹션을 참조하세요. AWS CLI

### `create-field-level-encryption-profile`

다음 코드 예시에서는 `create-field-level-encryption-profile`을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudFront 필드 수준 암호화 프로파일을 생성하려면

다음 예제에서는 라는 JSON 파일에 파라미터를 제공하여 필드 수준 암호화 프로파일을 생성합니다`fle-profile-config.json`. 필드 수준 암호화 프로파일을 생성하려면 먼저 CloudFront 퍼블릭 키가 있어야 합니다. CloudFront 퍼블릭 키를 생성하려면 명령을 참조하세요 create-public-key.

CloudFront 필드 수준 암호화에 대한 자세한 내용은 Amazon CloudFront 개발자 안내서의 필드 수준 암호화를 사용하여 민감한 데이터 보호를 참조하세요.

```
aws cloudfront create-field-level-encryption-profile \
    --field-level-encryption-profile-config file://fle-profile-config.json
```

파일은 현재 폴더의 JSON 문서`fle-profile-config.json`로, 다음을 포함합니다.

```
{
    "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"
                    ]
                }
            }
        ]
    }
}
```

출력:

```
{
    "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"
                            ]
                        }
                    }
                ]
            }
        }
    }
}
```

- 자세한 API 내용은 명령 참조CreateFieldLevelEncryptionProfile의 섹션을 참조하세요. AWS CLI

### `create-invalidation`

다음 코드 예시에서는 `create-invalidation`을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudFront 배포에 대한 무효화를 생성하려면

다음 `create-invalidation` 예제에서는 지정된 CloudFront 배포에서 지정된 파일에 대해 무효화를 생성합니다.

```
aws cloudfront create-invalidation \
    --distribution-id EDFDVBD6EXAMPLE \
    --paths "/example-path/example-file.jpg" "/example-path/example-file2.png"
```

출력:

```
{
    "Location": "https://cloudfront.amazonaws.com/2019-03-26/distribution/
EDFDVBD6EXAMPLE/invalidation/I1JLWSDAP8FU89",
    "Invalidation": {
        "Id": "I1JLWSDAP8FU89",
        "Status": "InProgress",
        "CreateTime": "2019-12-05T18:24:51.407Z",
        "InvalidationBatch": {
            "Paths": {
                "Quantity": 2,
                "Items": [
                    "/example-path/example-file2.png",
                    "/example-path/example-file.jpg"
```

```
                ]
            },
            "CallerReference": "cli-1575570291-670203"
        }
    }
}
```

이전 예제에서 는 AWS CLI 무작위 를 자동으로 생성했습니다`CallerReference`. 자체 를 지정`CallerReference`하거나 무효화 파라미터를 명령줄 인수로 전달하지 않으려면 JSON 파일을 사용할 수 있습니다. 다음 예제에서는 라는 파일에 무효화 파라미터를 제공하여 두 JSON 파일에 대한 무효화를 생성합니다`inv-batch.json`.

```
aws cloudfront create-invalidation \
    --distribution-id EDFDVBD6EXAMPLE \
    --invalidation-batch file://inv-batch.json
```

`inv-batch.json`의 콘텐츠:

```
{
    "Paths": {
        "Quantity": 2,
        "Items": [
            "/example-path/example-file.jpg",
            "/example-path/example-file2.png"
        ]
    },
    "CallerReference": "cli-example"
}
```

출력:

```
{
    "Location": "https://cloudfront.amazonaws.com/2019-03-26/distribution/
EDFDVBD6EXAMPLE/invalidation/I2J0I21PCUYOIK",
    "Invalidation": {
        "Id": "I2J0I21PCUYOIK",
        "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"
        }
    }
}
```

- 자세한 API 내용은 명령 참조CreateInvalidation의 섹션을 참조하세요. AWS CLI

### `create-public-key`

다음 코드 예시에서는 `create-public-key`을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudFront 퍼블릭 키를 생성하려면

다음 예제에서는 라는 JSON 파일에 파라미터를 제공하여 CloudFront 퍼블릭 키를 생성합니다pub-key-config.json. 이 명령을 사용하려면 먼저 PEM인코딩된 퍼블릭 키가 있어야 합니다. 자세한 내용은 Amazon CloudFront 개발자 안내서의 RSA 키 페어 생성을 참조하세요.

```
aws cloudfront create-public-key \
    --public-key-config file://pub-key-config.json
```

파일은 현재 폴더의 JSON 문서pub-key-config.json로 다음을 포함합니다. 퍼블릭 키는 PEM 형식으로 인코딩됩니다.

```
{
    "CallerReference": "cli-example",
    "Name": "ExampleKey",
    "EncodedKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxPMbCA2Ks0lnd7IR+3pw
\nwd3H/7jPGwj8bLUmore7bX+oeGpZ6QmLAe/1UOWcmZX2u70dYcSIzB1ofZtcn4cJ
\nenHBAzO3ohBY/L1tQGJfS2A+omnN6H16VZE1JCK8XSJyfze7MDLcUyHZETdxuvRb
\nA9X343/vMAuQPnhinFJ8Wdy8YBXSPpy7r95ylUQd9LfYTBzVZYG2tSesplcOkjM3\n2Uu
+oMWxQAw1NINnSLPinMVsutJy6ZqlV3McWNWe4T+STGtWhrPNqJEn45sIcCx4\nq
+kGZ2NQ0FyIyT2eiLKOX5Rgb/a36E/aMk4VoDsaenBQgG7WLTnstb9sr7MIhS6A\nrwIDAQAB\n-----END
 PUBLIC KEY-----\n",
    "Comment": "example public key"
}
```

출력:

```
{
    "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/1UOWcmZX2u70dYcSIzB1ofZtcn4cJ
\nenHBAzO3ohBY/L1tQGJfS2A+omnN6H16VZE1JCK8XSJyfze7MDLcUyHZETdxuvRb
\nA9X343/vMAuQPnhinFJ8Wdy8YBXSPpy7r95ylUQd9LfYTBzVZYG2tSesplcOkjM3\n2Uu
+oMWxQAw1NINnSLPinMVsutJy6ZqlV3McWNWe4T+STGtWhrPNqJEn45sIcCx4\nq
+kGZ2NQ0FyIyT2eiLKOX5Rgb/a36E/aMk4VoDsaenBQgG7WLTnstb9sr7MIhS6A\nrwIDAQAB\n-----END
 PUBLIC KEY-----\n",
            "Comment": "example public key"
        }
    }
}
```

- 자세한 API 내용은 명령 참조CreatePublicKey의 섹션을 참조하세요. AWS CLI

### `delete-cloud-front-origin-access-identity`

다음 코드 예시에서는 `delete-cloud-front-origin-access-identity`을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudFront 오리진 액세스 ID를 삭제하려면

다음 예제에서는 ID가 인 오리진 액세스 자격 증명(OAI)을 삭제합니다`E74FTE3AEXAMPLE`. 를 삭제하려면 OAI의 ID와 가 있어야 OAI합니다`ETag`. OAI ID는 create-cloud-front-origin-access-identity 및 list-cloud-front-origin-access-identities 명령의 출력에 반환됩니다. 를 가져오려면 get-cloud-front-origin-access-identity 또는 get-cloud-front-origin-access-identity-config command를 `ETag`사용합니다. `--if-match` 옵션을 사용하여 의 OAI를 제공합니다`ETag`.

```
aws cloudfront delete-cloud-front-origin-access-identity \
    --id E74FTE3AEXAMPLE \
    --if-match E2QWRUHEXAMPLE
```

이 명령이 제대로 실행되면 출력이 표시되지 않습니다.

- 자세한 API 내용은 명령 참조DeleteCloudFrontOriginAccessIdentity의 섹션을 참조하세요. AWS CLI

### delete-distribution

다음 코드 예시에서는 delete-distribution을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudFront 배포를 삭제하려면

다음 예제에서는 ID가 인 CloudFront 배포를 삭제합니다EDFDVBD6EXAMPLE. 배포를 삭제하려면 먼저 배포를 비활성화해야 합니다. 배포를 비활성화하려면 update-distribution 명령을 사용하세요. 자세한 정보는 update-distribution 섹션을 참조하세요.

배포가 비활성화된 경우 이를 삭제할 수 있습니다. 배포를 삭제하려면 배포의 ETag를 제공하는 --if-match 옵션을 사용해야 합니다. 를 가져오려면 get-distribution 또는 get-distribution-config 명령을 ETag사용합니다.

```
aws cloudfront delete-distribution \
    --id EDFDVBD6EXAMPLE \
    --if-match E2QWRUHEXAMPLE
```

이 명령이 제대로 실행되면 출력이 표시되지 않습니다.

- 자세한 API 내용은 명령 참조DeleteDistribution의 섹션을 참조하세요. AWS CLI

### delete-field-level-encryption-config

다음 코드 예시에서는 delete-field-level-encryption-config을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudFront 필드 수준 암호화 구성을 삭제하려면

다음 예제에서는 ID 를 사용하여 CloudFront 필드 수준 암호화 구성을 삭제합니다C3KM2WVD605UAY. 필드 수준 암호화 구성을 삭제하려면 ID와 이 있어야 합니다ETag. ID는 create-field-level-encryption-config 및 list-field-level-encryption-config 명령의 출력으로 반환됩니다. 를 가져오려면 get-field-level-encryption 또는 get-field-level-encryption-config 명령을 ETag사용합니다. --if-match 옵션을 사용하여 구성의 를 제공합니다ETag.

```
aws cloudfront delete-field-level-encryption-config \
    --id C3KM2WVD605UAY \
    --if-match E26M4BIAV81ZF6
```

이 명령이 제대로 실행되면 출력이 표시되지 않습니다.

- 자세한 API 내용은 명령 참조DeleteFieldLevelEncryptionConfig의 섹션을 참조하세요. AWS CLI

### delete-field-level-encryption-profile

다음 코드 예시에서는 delete-field-level-encryption-profile을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudFront 필드 수준 암호화 프로파일을 삭제하려면

다음 예제에서는 ID가 인 CloudFront 필드 수준 암호화 프로파일을 삭제합니다PPK0UOSIF5WSV. 필드 수준 암호화 프로파일을 삭제하려면 ID와 이 있어야 합니다ETag. ID는 create-field-level-encryption-profile 및 list-field-level-encryption-profiles 명령의 출력에 반환됩니다. 를 가져오려면 get-field-level-encryption-profile 또는 get-field-level-encryption-profile-config 명령을 ETag사용합니다. --if-match 옵션을 사용하여 프로필의 를 제공합니다ETag.

```
aws cloudfront delete-field-level-encryption-profile \
    --id PPK0UOSIF5WSV \
    --if-match EJETYFJ9CL66D
```

이 명령이 제대로 실행되면 출력이 표시되지 않습니다.

- 자세한 API 내용은 명령 참조DeleteFieldLevelEncryptionProfile의 섹션을 참조하세요. AWS CLI

### delete-public-key

다음 코드 예시에서는 delete-public-key을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudFront 퍼블릭 키를 삭제하려면

다음 예제에서는 ID가 인 CloudFront 퍼블릭 키를 삭제합니다KDFB19YGCR002. 퍼블릭 키를 삭제하려면 ID와 가 있어야 합니다ETag. create-public-key 및 list-public-keys 명령의 출력에 ID가 반환됩니다. 를 가져오려면 get-public-key 또는 get-public-key-config 명령을 ETag사용합니다. `--if-match` 옵션을 사용하여 퍼블릭 키의 를 제공합니다ETag.

```
aws cloudfront delete-public-key \
    --id KDFB19YGCR002 \
    --if-match E2QWRUHEXAMPLE
```

이 명령이 제대로 실행되면 출력이 표시되지 않습니다.

- 자세한 API 내용은 명령 참조DeletePublicKey의 섹션을 참조하세요. AWS CLI

## get-cloud-front-origin-access-identity-config

다음 코드 예시에서는 `get-cloud-front-origin-access-identity-config`을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudFront 오리진 액세스 자격 증명 구성을 가져오려면

다음 예제에서는 를 E74FTE3AEXAMPLE포함하여 ID 를 사용하여 CloudFront 오리진 액세스 자격 증명(OAI)에 대한 메타데이터를 가져옵니다ETag. OAI ID는 create-cloud-front-origin-access-identity 및 list-cloud-front-origin-access-identities 명령의 출력으로 반환됩니다.

```
aws cloudfront get-cloud-front-origin-access-identity-config --id E74FTE3AEXAMPLE
```

출력:

```
{
    "ETag": "E2QWRUHEXAMPLE",
    "CloudFrontOriginAccessIdentityConfig": {
        "CallerReference": "cli-example",
        "Comment": "Example OAI"
    }
```

```
}
```

- 자세한 API 내용은 명령 참조GetCloudFrontOriginAccessIdentityConfig의 섹션을 참조하세요. AWS CLI

### get-cloud-front-origin-access-identity

다음 코드 예시에서는 get-cloud-front-origin-access-identity을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudFront 오리진 액세스 자격 증명을 가져오려면

다음 예제에서는 ETag 및 연결된 S3 정식 ID를 E74FTE3AEXAMPLE포함하여 ID 를 사용하여 CloudFront 오리진 액세스 자격 증명(OAI)을 가져옵니다. OAI ID는 create-cloud-front-origin-access-identity 및 list-cloud-front-origin-access-identities 명령의 출력으로 반환됩니다.

```
aws cloudfront get-cloud-front-origin-access-identity --id E74FTE3AEXAMPLE
```

출력:

```
{
    "ETag": "E2QWRUHEXAMPLE",
    "CloudFrontOriginAccessIdentity": {
        "Id": "E74FTE3AEXAMPLE",
        "S3CanonicalUserId":
 "cd13868f797c227fbea2830611a26fe0a21ba1b826ab4bed9b7771c9aEXAMPLE",
        "CloudFrontOriginAccessIdentityConfig": {
            "CallerReference": "cli-example",
            "Comment": "Example OAI"
        }
    }
}
```

- 자세한 API 내용은 명령 참조GetCloudFrontOriginAccessIdentity의 섹션을 참조하세요. AWS CLI

### get-distribution-config

다음 코드 예시에서는 get-distribution-config을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudFront 배포 구성을 가져오려면

다음 예제에서는 를 EDFDVBD6EXAMPLE포함하여 ID 를 사용하여 CloudFront 배포에 대한 메타데이터를 가져옵니다ETag. 배포 ID는 create-distribution 및 list-distributions 명령에서 반환됩니다.

```
aws cloudfront get-distribution-config --id EDFDVBD6EXAMPLE
```

출력:

```
{
    "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
    }
}
```

- 자세한 API 내용은 명령 참조GetDistributionConfig의 섹션을 참조하세요. AWS CLI

### `get-distribution`

다음 코드 예시에서는 `get-distribution`을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudFront 배포를 가져오려면

다음 예제에서는 를 `EDFDVBD6EXAMPLE`포함하여 ID 를 사용하여 CloudFront 배포를 가져옵니다ETag. 배포 ID는 create-distribution 및 list-distributions 명령에서 반환됩니다.

```
aws cloudfront get-distribution --id EDFDVBD6EXAMPLE
```

출력:

```
{
    "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
        }
    }
}
```

- 자세한 API 내용은 명령 참조GetDistribution의 섹션을 참조하세요. AWS CLI

### `get-field-level-encryption-config`

다음 코드 예시에서는 `get-field-level-encryption-config`을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudFront 필드 수준 암호화 구성에 대한 메타데이터를 가져오려면

다음 예제에서는 를 `C3KM2WVD605UAY`포함하여 ID가 인 CloudFront 필드 수준 암호화 구성에 대한 메타데이터를 가져옵니다`ETag`.

```
aws cloudfront get-field-level-encryption-config --id C3KM2WVD605UAY
```

출력:

```
{
    "ETag": "E2P4Z4VU7TY5SG",
    "FieldLevelEncryptionConfig": {
        "CallerReference": "cli-example",
        "Comment": "Example FLE configuration",
        "QueryArgProfileConfig": {
            "ForwardWhenQueryArgProfileIsUnknown": true,
            "QueryArgProfiles": {
                "Quantity": 0,
                "Items": []
            }
        },
        "ContentTypeProfileConfig": {
            "ForwardWhenContentTypeIsUnknown": true,
            "ContentTypeProfiles": {
                "Quantity": 1,
                "Items": [
                    {
                        "Format": "URLEncoded",
                        "ProfileId": "P280MFCLSYOCVU",
                        "ContentType": "application/x-www-form-urlencoded"
                    }
                ]
            }
        }
    }
}
```

- 자세한 API 내용은 명령 참조GetFieldLevelEncryptionConfig의 섹션을 참조하세요. AWS CLI

### get-field-level-encryption-profile-config

다음 코드 예시에서는 get-field-level-encryption-profile-config을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudFront 필드 수준 암호화 프로필 구성을 가져오려면

다음 예제에서는 를 포함하여 IDPPK0UOSIF5WSV가 인 CloudFront 필드 수준 암호화 프로파일에 대한 메타데이터를 가져옵니다ETag.

```
aws cloudfront get-field-level-encryption-profile-config --id PPK0UOSIF5WSV
```

출력:

```
{
    "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"
                        ]
                    }
                }
            ]
        }
    }
}
```

- 자세한 API 내용은 명령 참조GetFieldLevelEncryptionProfileConfig의 섹션을 참조하세요. AWS CLI

### get-field-level-encryption-profile

다음 코드 예시에서는 get-field-level-encryption-profile을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudFront 필드 수준 암호화 프로파일을 가져오려면

다음 예제에서는 를 포함하여 IDPPK0UOSIF5WSV가 인 CloudFront 필드 수준 암호화 프로파일을 가져옵니다ETag.

```
aws cloudfront get-field-level-encryption-profile --id PPK0UOSIF5WSV
```

출력:

```
{
    "ETag": "E1QQG65FS2L2GC",
    "FieldLevelEncryptionProfile": {
        "Id": "PPK0UOSIF5WSV",
        "LastModifiedTime": "2019-12-10T01:03:16.537Z",
        "FieldLevelEncryptionProfileConfig": {
            "Name": "ExampleFLEProfile",
            "CallerReference": "cli-example",
            "Comment": "FLE profile for AWS CLI example",
            "EncryptionEntities": {
                "Quantity": 1,
                "Items": [
                    {
                        "PublicKeyId": "K2K8NC4HVFE3M0",
                        "ProviderId": "ExampleFLEProvider",
                        "FieldPatterns": {
                            "Quantity": 1,
                            "Items": [
                                "ExampleSensitiveField"
                            ]
                        }
                    }
                ]
            }
        }
    }
}
```

- 자세한 API 내용은 명령 참조GetFieldLevelEncryptionProfile의 섹션을 참조하세요. AWS CLI

### `get-field-level-encryption`

다음 코드 예시에서는 `get-field-level-encryption`을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudFront 필드 수준 암호화 구성을 가져오려면

다음 예제에서는 를 C3KM2WVD605UAY포함하여 ID가 인 CloudFront 필드 수준 암호화 구성을 가져옵니다ETag.

```
aws cloudfront get-field-level-encryption --id C3KM2WVD605UAY
```

출력:

```
{
    "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"
                        }
                    ]
                }
            }
        }
    }
}
```

- 자세한 API 내용은 명령 참조GetFieldLevelEncryption의 섹션을 참조하세요. AWS CLI

### get-invalidation

다음 코드 예시에서는 get-invalidation을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudFront 무효화를 가져오려면

다음 예제에서는 ID 를 사용하는 I2J0I21PCUYOIK CloudFront 배포의 ID 를 사용하여 무효화를 가져옵니다EDFDVBD6EXAMPLE.

```
aws cloudfront get-invalidation --id I2J0I21PCUYOIK --distribution-
id EDFDVBD6EXAMPLE
```

출력:

```
{
    "Invalidation": {
        "Status": "Completed",
        "InvalidationBatch": {
            "Paths": {
                "Items": [
                    "/example-path/example-file.jpg",
                    "/example-path/example-file-2.jpg"
                ],
                "Quantity": 2
            },
            "CallerReference": "cli-example"
        },
        "Id": "I2J0I21PCUYOIK",
        "CreateTime": "2019-12-05T18:40:49.413Z"
    }
}
```

- 자세한 API 내용은 명령 참조GetInvalidation의 섹션을 참조하세요. AWS CLI

### get-public-key-config

다음 코드 예시에서는 get-public-key-config을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudFront 퍼블릭 키 구성을 가져오려면

다음 예제에서는 를 KDFB19YGCR002포함하여 ID가 인 CloudFront 퍼블릭 키에 대한 메타데이터를 가져옵니다ETag. create-public-key 및 list-public-keys 명령에서 퍼블릭 키 ID가 반환됩니다.

```
aws cloudfront get-public-key-config --id KDFB19YGCR002
```

출력:

```
{
    "ETag": "E2QWRUHEXAMPLE",
    "PublicKeyConfig": {
        "CallerReference": "cli-example",
        "Name": "ExampleKey",
        "EncodedKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxPMbCA2Ks0lnd7IR+3pw
\nwd3H/7jPGwj8bLUmore7bX+oeGpZ6QmLAe/1UOWcmZX2u70dYcSIzB1ofZtcn4cJ
\nenHBAzO3ohBY/L1tQGJfS2A+omnN6H16VZE1JCK8XSJyfze7MDLcUyHZETdxuvRb
\nA9X343/vMAuQPnhinFJ8Wdy8YBXSPpy7r95ylUQd9LfYTBzVZYG2tSesplcOkjM3\n2Uu
+oMWxQAw1NINnSLPinMVsutJy6ZqlV3McWNWe4T+STGtWhrPNqJEn45sIcCx4\nq
+kGZ2NQ0FyIyT2eiLKOX5Rgb/a36E/aMk4VoDsaenBQgG7WLTnstb9sr7MIhS6A\nrwIDAQAB\n-----END
 PUBLIC KEY-----\n",
        "Comment": "example public key"
    }
}
```

- 자세한 API 내용은 명령 참조GetPublicKeyConfig의 섹션을 참조하세요. AWS CLI

### get-public-key

다음 코드 예시에서는 get-public-key을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudFront 퍼블릭 키를 가져오려면

다음 예제에서는 를 KDFB19YGCR002포함하여 ID가 인 CloudFront 퍼블릭 키를 가져옵니다ETag. create-public-key 및 list-public-keys 명령에서 퍼블릭 키 ID가 반환됩니다.

```
aws cloudfront get-public-key --id KDFB19YGCR002
```

출력:

```
{
    "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/1UOWcmZX2u70dYcSIzB1ofZtcn4cJ
\nenHBAzO3ohBY/L1tQGJfS2A+omnN6H16VZE1JCK8XSJyfze7MDLcUyHZETdxuvRb
\nA9X343/vMAuQPnhinFJ8Wdy8YBXSPpy7r95ylUQd9LfYTBzVZYG2tSesplcOkjM3\n2Uu
+oMWxQAw1NINnSLPinMVsutJy6ZqlV3McWNWe4T+STGtWhrPNqJEn45sIcCx4\nq
+kGZ2NQ0FyIyT2eiLKOX5Rgb/a36E/aMk4VoDsaenBQgG7WLTnstb9sr7MIhS6A\nrwIDAQAB\n-----END
 PUBLIC KEY-----\n",
            "Comment": "example public key"
        }
    }
}
```

- 자세한 API 내용은 명령 참조GetPublicKey의 섹션을 참조하세요. AWS CLI

### `list-cloud-front-origin-access-identities`

다음 코드 예시에서는 `list-cloud-front-origin-access-identities`을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudFront 오리진 액세스 ID를 나열하려면

다음 예제에서는 AWS 계정의 CloudFront 오리진 액세스 자격 증명(OAIs) 목록을 가져옵니다.

```
aws cloudfront list-cloud-front-origin-access-identities
```

출력:

```
{
    "CloudFrontOriginAccessIdentityList": {
        "Items": [
            {
```

```
                "Id": "E74FTE3AEXAMPLE",
                "S3CanonicalUserId":
 "cd13868f797c227fbea2830611a26fe0a21ba1b826ab4bed9b7771c9aEXAMPLE",
                "Comment": "Example OAI"
            },
            {
                "Id": "EH1HDMBEXAMPLE",
                "S3CanonicalUserId":
 "1489f6f2e6faacaae7ff64c4c3e6956c24f78788abfc1718c3527c263bf7a17EXAMPLE",
                "Comment": "Test OAI"
            },
            {
                "Id": "E2X2C9TEXAMPLE",
                "S3CanonicalUserId":
 "cbfeebb915a64749f9be546a45b3fcfd3a31c779673c13c4dd460911ae402c2EXAMPLE",
                "Comment": "Example OAI #2"
            }
        ]
    }
}
```

- 자세한 API 내용은 명령 참조ListCloudFrontOriginAccessIdentities의 섹션을 참조하세요. AWS CLI

### `list-distributions`

다음 코드 예시에서는 `list-distributions`을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudFront 배포를 나열하려면

다음 예제에서는 AWS 계정의 CloudFront 배포 목록을 가져옵니다.

```
aws cloudfront list-distributions
```

출력:

```
{
    "DistributionList": {
        "Items": [
            {
```

```
                "Id": "EMLARXS9EXAMPLE",
                "ARN": "arn:aws:cloudfront::123456789012:distribution/
EMLARXS9EXAMPLE",
                "Status": "InProgress",
                "LastModifiedTime": "2019-11-22T00:55:15.705Z",
                "InProgressInvalidationBatches": 0,
                "DomainName": "d111111abcdef8.cloudfront.net",
                "ActiveTrustedSigners": {
                    "Enabled": false,
                    "Quantity": 0
                },
                "DistributionConfig": {
                    "CallerReference": "cli-example",
                    "Aliases": {
                        "Quantity": 0
                    },
                    "DefaultRootObject": "index.html",
                    "Origins": {
                        "Quantity": 1,
                        "Items": [
                            {
                                "Id": "awsexamplebucket.s3.amazonaws.com-cli-
example",
                                "DomainName": "awsexamplebucket.s3.amazonaws.com",
                                "OriginPath": "",
                                "CustomHeaders": {
                                    "Quantity": 0
                                },
                                "S3OriginConfig": {
                                    "OriginAccessIdentity": ""
                                }
                            }
                        ]
                    },
                    "OriginGroups": {
                        "Quantity": 0
                    },
                    "DefaultCacheBehavior": {
                        "TargetOriginId": "awsexamplebucket.s3.amazonaws.com-cli-
example",
                        "ForwardedValues": {
                            "QueryString": false,
                            "Cookies": {
                                "Forward": "none"
```

```
                    },
                    "Headers": {
                        "Quantity": 0
                    },
                    "QueryStringCacheKeys": {
                        "Quantity": 0
                    }
                },
                "TrustedSigners": {
                    "Enabled": false,
                    "Quantity": 0
                },
                "ViewerProtocolPolicy": "allow-all",
                "MinTTL": 0,
                "AllowedMethods": {
                    "Quantity": 2,
                    "Items": [
                        "HEAD",
                        "GET"
                    ],
                    "CachedMethods": {
                        "Quantity": 2,
                        "Items": [
                            "HEAD",
                            "GET"
                        ]
                    }
                },
                "SmoothStreaming": false,
                "DefaultTTL": 86400,
                "MaxTTL": 31536000,
                "Compress": false,
                "LambdaFunctionAssociations": {
                    "Quantity": 0
                },
                "FieldLevelEncryptionId": ""
            },
            "CacheBehaviors": {
                "Quantity": 0
            },
            "CustomErrorResponses": {
                "Quantity": 0
            },
            "Comment": "",
```

```
                    "Logging": {
                        "Enabled": false,
                        "IncludeCookies": false,
                        "Bucket": "",
                        "Prefix": ""
                    },
                    "PriceClass": "PriceClass_All",
                    "Enabled": true,
                    "ViewerCertificate": {
                        "CloudFrontDefaultCertificate": true,
                        "MinimumProtocolVersion": "TLSv1",
                        "CertificateSource": "cloudfront"
                    },
                    "Restrictions": {
                        "GeoRestriction": {
                            "RestrictionType": "none",
                            "Quantity": 0
                        }
                    },
                    "WebACLId": "",
                    "HttpVersion": "http2",
                    "IsIPV6Enabled": true
                }
            },
            {
                "Id": "EDFDVBD6EXAMPLE",
                "ARN": "arn:aws:cloudfront::123456789012:distribution/
EDFDVBD6EXAMPLE",
                "Status": "InProgress",
                "LastModifiedTime": "2019-12-04T23:35:41.433Z",
                "InProgressInvalidationBatches": 0,
                "DomainName": "d930174dauwrn8.cloudfront.net",
                "ActiveTrustedSigners": {
                    "Enabled": false,
                    "Quantity": 0
                },
                "DistributionConfig": {
                    "CallerReference": "cli-example",
                    "Aliases": {
                        "Quantity": 0
                    },
                    "DefaultRootObject": "index.html",
                    "Origins": {
                        "Quantity": 1,
```

```
                        "Items": [
                            {
                                "Id": "awsexamplebucket1.s3.amazonaws.com-cli-
example",
                                "DomainName": "awsexamplebucket1.s3.amazonaws.com",
                                "OriginPath": "",
                                "CustomHeaders": {
                                    "Quantity": 0
                                },
                                "S3OriginConfig": {
                                    "OriginAccessIdentity": ""
                                }
                            }
                        ]
                    },
                    "OriginGroups": {
                        "Quantity": 0
                    },
                    "DefaultCacheBehavior": {
                        "TargetOriginId": "awsexamplebucket1.s3.amazonaws.com-cli-
example",
                        "ForwardedValues": {
                            "QueryString": false,
                            "Cookies": {
                                "Forward": "none"
                            },
                            "Headers": {
                                "Quantity": 0
                            },
                            "QueryStringCacheKeys": {
                                "Quantity": 0
                            }
                        },
                        "TrustedSigners": {
                            "Enabled": false,
                            "Quantity": 0
                        },
                        "ViewerProtocolPolicy": "allow-all",
                        "MinTTL": 0,
                        "AllowedMethods": {
                            "Quantity": 2,
                            "Items": [
                                "HEAD",
                                "GET"
```

```
                ],
                "CachedMethods": {
                    "Quantity": 2,
                    "Items": [
                        "HEAD",
                        "GET"
                    ]
                }
            },
            "SmoothStreaming": false,
            "DefaultTTL": 86400,
            "MaxTTL": 31536000,
            "Compress": false,
            "LambdaFunctionAssociations": {
                "Quantity": 0
            },
            "FieldLevelEncryptionId": ""
        },
        "CacheBehaviors": {
            "Quantity": 0
        },
        "CustomErrorResponses": {
            "Quantity": 0
        },
        "Comment": "",
        "Logging": {
            "Enabled": false,
            "IncludeCookies": false,
            "Bucket": "",
            "Prefix": ""
        },
        "PriceClass": "PriceClass_All",
        "Enabled": true,
        "ViewerCertificate": {
            "CloudFrontDefaultCertificate": true,
            "MinimumProtocolVersion": "TLSv1",
            "CertificateSource": "cloudfront"
        },
        "Restrictions": {
            "GeoRestriction": {
                "RestrictionType": "none",
                "Quantity": 0
            }
        },
```

```
                    "WebACLId": "",
                    "HttpVersion": "http2",
                    "IsIPV6Enabled": true
                }
            },
            {
                "Id": "E1X5IZQEXAMPLE",
                "ARN": "arn:aws:cloudfront::123456789012:distribution/
E1X5IZQEXAMPLE",
                "Status": "Deployed",
                "LastModifiedTime": "2019-11-06T21:31:48.864Z",
                "DomainName": "d2e04y12345678.cloudfront.net",
                "Aliases": {
                    "Quantity": 0
                },
                "Origins": {
                    "Quantity": 1,
                    "Items": [
                        {
                            "Id": "awsexamplebucket2",
                            "DomainName": "awsexamplebucket2.s3.us-
west-2.amazonaws.com",
                            "OriginPath": "",
                            "CustomHeaders": {
                                "Quantity": 0
                            },
                            "S3OriginConfig": {
                                "OriginAccessIdentity": ""
                            }
                        }
                    ]
                },
                "OriginGroups": {
                    "Quantity": 0
                },
                "DefaultCacheBehavior": {
                    "TargetOriginId": "awsexamplebucket2",
                    "ForwardedValues": {
                        "QueryString": false,
                        "Cookies": {
                            "Forward": "none"
                        },
                        "Headers": {
                            "Quantity": 0
```

```
                    },
                    "QueryStringCacheKeys": {
                        "Quantity": 0
                    }
                },
                "TrustedSigners": {
                    "Enabled": false,
                    "Quantity": 0
                },
                "ViewerProtocolPolicy": "allow-all",
                "MinTTL": 0,
                "AllowedMethods": {
                    "Quantity": 2,
                    "Items": [
                        "HEAD",
                        "GET"
                    ],
                    "CachedMethods": {
                        "Quantity": 2,
                        "Items": [
                            "HEAD",
                            "GET"
                        ]
                    }
                },
                "SmoothStreaming": false,
                "DefaultTTL": 86400,
                "MaxTTL": 31536000,
                "Compress": false,
                "LambdaFunctionAssociations": {
                    "Quantity": 0
                },
                "FieldLevelEncryptionId": ""
            },
            "CacheBehaviors": {
                "Quantity": 0
            },
            "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
            }
        ]
    }
}
```

- 자세한 API 내용은 명령 참조ListDistributions의 섹션을 참조하세요. AWS CLI

### `list-field-level-encryption-configs`

다음 코드 예시에서는 `list-field-level-encryption-configs`을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudFront 필드 수준 암호화 구성을 나열하려면

다음 예제에서는 AWS 계정의 CloudFront 필드 수준 암호화 구성 목록을 가져옵니다.

```
aws cloudfront list-field-level-encryption-configs
```

출력:

```
{
    "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"
                            }
                        ]
                    }
                }
            }
        ]
    }
}
```

- 자세한 API 내용은 명령 참조ListFieldLevelEncryptionConfigs의 섹션을 참조하세요. AWS CLI

### `list-field-level-encryption-profiles`

다음 코드 예시에서는 `list-field-level-encryption-profiles`을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudFront 필드 수준 암호화 프로파일을 나열하려면

다음 예제에서는 AWS 계정의 CloudFront 필드 수준 암호화 프로파일 목록을 가져옵니다.

```
aws cloudfront list-field-level-encryption-profiles
```

출력:

```
{
    "FieldLevelEncryptionProfileList": {
        "MaxItems": 100,
        "Quantity": 2,
        "Items": [
            {
                "Id": "P280MFCLSYOCVU",
                "LastModifiedTime": "2019-12-05T01:05:39.896Z",
                "Name": "ExampleFLEProfile",
                "EncryptionEntities": {
                    "Quantity": 1,
                    "Items": [
                        {
                            "PublicKeyId": "K2K8NC4HVFE3M0",
                            "ProviderId": "ExampleFLEProvider",
                            "FieldPatterns": {
                                "Quantity": 1,
                                "Items": [
                                    "ExampleSensitiveField"
                                ]
                            }
                        }
                    ]
                },
                "Comment": "FLE profile for AWS CLI example"
            },
            {
                "Id": "PPK0UOSIF5WSV",
                "LastModifiedTime": "2019-12-10T01:03:16.537Z",
                "Name": "ExampleFLEProfile2",
                "EncryptionEntities": {
                    "Quantity": 1,
                    "Items": [
                        {
                            "PublicKeyId": "K2ABC10EXAMPLE",
                            "ProviderId": "ExampleFLEProvider2",
                            "FieldPatterns": {
                                "Quantity": 1,
                                "Items": [
                                    "ExampleSensitiveField2"
                                ]
                            }
```

```
                        }
                    ]
                },
                "Comment": "FLE profile #2 for AWS CLI example"
            }
        ]
    }
}
```

- 자세한 API 내용은 명령 참조ListFieldLevelEncryptionProfiles의 섹션을 참조하세요. AWS CLI

### `list-invalidations`

다음 코드 예시에서는 `list-invalidations`을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudFront 무효화를 나열하려면

다음 예제에서는 ID 를 사용하여 배포에 대한 CloudFront 무효화 목록을 가져옵니다EDFDVBD6EXAMPLE.

```
aws cloudfront list-invalidations --distribution-id EDFDVBD6EXAMPLE
```

출력:

```
{
    "InvalidationList": {
        "Marker": "",
        "Items": [
            {
                "Status": "Completed",
                "Id": "YNY2LI2BVJ4NJU",
                "CreateTime": "2019-08-31T21:15:52.042Z"
            }
        ],
        "IsTruncated": false,
        "MaxItems": 100,
        "Quantity": 1
    }
}
```

- 자세한 API 내용은 명령 참조ListInvalidations의 섹션을 참조하세요. AWS CLI

### `list-public-keys`

다음 코드 예시에서는 `list-public-keys`을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudFront 퍼블릭 키를 나열하려면

다음 예제에서는 AWS 계정의 CloudFront 퍼블릭 키 목록을 가져옵니다.

```
aws cloudfront list-public-keys
```

출력:

```
{
    "PublicKeyList": {
        "MaxItems": 100,
        "Quantity": 2,
        "Items": [
            {
                "Id": "K2K8NC4HVFE3M0",
                "Name": "ExampleKey",
                "CreatedTime": "2019-12-05T01:04:28.818Z",
                "EncodedKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxPMbCA2Ks0lnd7IR+3pw
\nwd3H/7jPGwj8bLUmore7bX+oeGpZ6QmLAe/1UOWcmZX2u70dYcSIzB1ofZtcn4cJ
\nenHBAzO3ohBY/L1tQGJfS2A+omnN6H16VZE1JCK8XSJyfze7MDLcUyHZETdxuvRb
\nA9X343/vMAuQPnhinFJ8Wdy8YBXSPpy7r95ylUQd9LfYTBzVZYG2tSesplcOkjM3\n2Uu
+oMWxQAw1NINnSLPinMVsutJy6ZqlV3McWNWe4T+STGtWhrPNqJEn45sIcCx4\nq
+kGZ2NQ0FyIyT2eiLKOX5Rgb/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-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEApOCAg88A8+f4dujn9Izt
\n26LxtgAkn2opGgo/NKpMiaisyw5qlg3f1gol7FV6pYNl78iJg3EO8JBbwtlH
```

```
+cR9\nLGSf60NDeVhm76Oc39Np/vWgOdsGQcRbi9WmKZeSODqjQGzVZWqPmito3FzWVk6b
\nfVY5N36U/RdbVAJm95Km+qaMYlbIdF40t72bi3IkKYV5hlB2XoDjlQ9F6ajQKyTB
\nMHa3SN8q+3ZjQ4sJJ7D1V6r4wR8jDcFVD5NckWJmmgIVnkOQM37NYeoDnkaOuTpu\nha/
+3b8tOb2z3LBVHPkp85zJRAOXacSwf5rZtPYKBNFsixTa2n55k2r218mOkMC4\nUwIDAQAB\n-----END
 PUBLIC KEY-----",
                "Comment": "example public key #2"
            }
        ]
    }
}
```

- 자세한 API 내용은 명령 참조ListPublicKeys의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudFront 배포에 대한 태그를 나열하려면

다음 예제에서는 CloudFront 배포에 대한 태그 목록을 가져옵니다.

```
aws cloudfront list-tags-for-resource \
    --resource arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE
```

출력:

```
{
    "Tags": {
        "Items": [
            {
                "Key": "DateCreated",
                "Value": "2019-12-04"
            },
            {
                "Key": "Name",
                "Value": "Example name"
            },
            {
                "Key": "Project",
                "Value": "Example project"
```

```
            }
        ]
    }
}
```

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### `sign`

다음 코드 예시에서는 `sign`을 사용하는 방법을 보여 줍니다.

AWS CLI

에 서명하려면 CloudFront URL

다음 예제는 에 서명합니다 CloudFront URL. 에 서명하려면 키 페어 ID( AWS 관리 콘솔에서 액세스 키 ID라고 함)와 신뢰할 수 있는 서명자의 키 페어의 프라이빗 CloudFront 키가 URL필요합니다. 서명된 에 대한 자세한 내용은 Amazon CloudFront 개발자 안내서의 서명된 쿠키 및 서명된 쿠키를 사용하여 프라이빗 콘텐츠 서비스를 URLs참조하세요. URLs

```
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
```

출력:

```
https://d111111abcdef8.cloudfront.net/private-content/private-
file.html?Expires=1577836800&Signature=nEXK7Kby47XKeZQKVc6pwkif6oZc-
JWSpDkH0UH7EBGGqvgurkecCbgL5VfUAXyLQuJxFwRQWscz-
owcq9KpmewCXrXQbPaJZNi9XSNwf4YKurPDQYaRQawKoeenH0GFteRf9ELK-
Bs3nljTLjtbgzIUt7QJNKXcWr8AuUYikzGdJ4-qzx6WnxXfH~fxg4-
GGl6l2kgCpXUB6Jx6K~Y3kpVOdzUPOIqFLHAnJojbhxqrVejomZZ2XrquDvNUCCIbePGnR3d24UPaLXG4FKOqNEaWDIB
GNvjRJxqWf93uMobeMOiVYahb-e0KItiQewGcm0eLZQ__&Key-Pair-Id=APKAEIBAERJR2EXAMPLE
```

- 자세한 API 내용은 로그인 AWS CLI 명령 참조를 참조하세요.

### `tag-resource`

다음 코드 예시에서는 `tag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudFront 배포에 태그를 지정하려면

다음 `tag-resource` 예제에서는 지정된 CloudFront 배포에 두 개의 태그를 추가합니다.

```
aws cloudfront tag-resource \
    --resource arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE \
    --tags 'Items=[{Key=Name,Value="Example name"},{Key=Project,Value="Example
 project"}]'
```

다음 예제와 같이 명령줄 인수를 사용하는 대신 JSON 파일에 태그를 제공할 수 있습니다.

```
aws cloudfront tag-resource \
    --resource arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE \
    --tags file://tags.json
```

`tags.json`의 콘텐츠:

```
{
    "Items": [
        {
            "Key": "Name",
            "Value": "Example name"
        },
        {
            "Key": "Project",
            "Value": "Example project"
        }
    ]
}
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### `untag-resource`

다음 코드 예시에서는 `untag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudFront 배포에서 태그를 제거하려면

다음 예제에서는 명령줄 인수를 사용하여 CloudFront 배포에서 두 개의 태그를 제거합니다.

```
aws cloudfront untag-resource \
    --resource arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE \
    --tag-keys Items=Name,Project
```

다음 예제와 같이 명령줄 인수를 사용하는 대신 JSON 파일에 태그 키를 제공할 수 있습니다.

```
aws cloudfront untag-resource \
    --resource arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE \
    --tag-keys file://tag-keys.json
```

파일은 현재 폴더의 JSON 문서tag-keys.json로, 다음을 포함합니다.

```
{
    "Items": [
        "Name",
        "Project"
    ]
}
```

이 명령이 제대로 실행되면 출력이 표시되지 않습니다.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-cloud-front-origin-access-identity

다음 코드 예시에서는 update-cloud-front-origin-access-identity을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudFront 오리진 액세스 자격 증명을 업데이트하려면

다음 예제에서는 오리진 액세스 자격 증명(OAI)을 ID 로 업데이트합니다E74FTE3AEXAMPLE. 업데이트할 수 있는 유일한 필드는 OAI의 입니다Comment.

를 업데이트하려면 OAI의 ID와 가 있어야 OAI합니다ETag. OAI ID는 create-cloud-front-origin-access-identity 및 list-cloud-front-origin-access-identities 명령의 출력에 반환됩니다. 를 가져오려면 get-cloud-front-origin-access-identity 또는 get-cloud-front-origin-access-identity-config command를 ETag사용합니다. --if-match 옵션을 사용하여 OAI의 를 제공합니다ETag.

```
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"
```

다음 예제와 같이 JSON 파일에 OAI 구성을 제공하여 동일한 작업을 수행할 수 있습니다.

```
aws cloudfront update-cloud-front-origin-access-identity \
    --id E74FTE3AEXAMPLE \
    --if-match E2QWRUHEXAMPLE \
    --cloud-front-origin-access-identity-config file://OAI-config.json
```

파일은 현재 디렉터리의 JSON 문서OAI-config.json로, 다음을 포함합니다.

```
{
    "CallerReference": "cli-example",
    "Comment": "Example OAI Updated"
}
```

명령줄 인수를 OAI 구성에 제공하든 JSON 파일을 제공하든 출력은 동일합니다.

```
{
    "ETag": "E9LHASXEXAMPLE",
    "CloudFrontOriginAccessIdentity": {
        "Id": "E74FTE3AEXAMPLE",
        "S3CanonicalUserId":
 "cd13868f797c227fbea2830611a26fe0a21ba1b826ab4bed9b7771c9aEXAMPLE",
        "CloudFrontOriginAccessIdentityConfig": {
            "CallerReference": "cli-example",
            "Comment": "Example OAI Updated"
        }
    }
}
```

- 자세한 API 내용은 명령 참조UpdateCloudFrontOriginAccessIdentity의 섹션을 참조하세요. AWS CLI

### update-distribution

다음 코드 예시에서는 update-distribution을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudFront 배포의 기본 루트 객체를 업데이트하려면

다음 예제에서는 ID 를 사용하여 CloudFront 배포에 `index.html` 대한 기본 루트 객체를 로 업데이트합니다`EDFDVBD6EXAMPLE`.

```
aws cloudfront update-distribution --id EDFDVBD6EXAMPLE \
    --default-root-object index.html
```

출력:

```
{
    "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
        }
    }
}
```

CloudFront 배포를 업데이트하려면

다음 예제에서는 라는 JSON 파일에 CloudFront 배포 구성을 EMLARXS9EXAMPLE 제공하여 ID 가 있는 배포를 비활성화합니다dist-config-disable.json. 배포를 업데이트하려면 배포의 ETag를 제공하는 --if-match 옵션을 사용해야 합니다. 를 가져오려면 get-distribution 또는 get-distribution-config 명령을 ETag사용합니다.

다음 예시를 사용하여 배포를 비활성화한 후 delete-distribution 명령을 사용하여 배포를 삭제할 수 있습니다.

```
aws cloudfront update-distribution \
    --id EMLARXS9EXAMPLE \
    --if-match E2QWRUHEXAMPLE \
    --distribution-config file://dist-config-disable.json
```

파일은 현재 폴더의 JSON 문서dist-config-disable.json로 다음을 포함합니다. Enabled 필드는 false로 설정됩니다.

```
{
    "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
}
```

출력:

```
{
```

```
    "ETag": "E9LHASXEXAMPLE",
    "Distribution": {
        "Id": "EMLARXS9EXAMPLE",
        "ARN": "arn:aws:cloudfront::123456789012:distribution/EMLARXS9EXAMPLE",
        "Status": "InProgress",
        "LastModifiedTime": "2019-12-06T18:32:35.553Z",
        "InProgressInvalidationBatches": 0,
        "DomainName": "d111111abcdef8.cloudfront.net",
        "ActiveTrustedSigners": {
            "Enabled": false,
            "Quantity": 0
        },
        "DistributionConfig": {
            "CallerReference": "cli-1574382155-496510",
            "Aliases": {
                "Quantity": 0
            },
            "DefaultRootObject": "index.html",
            "Origins": {
                "Quantity": 1,
                "Items": [
                    {
                        "Id": "awsexamplebucket.s3.amazonaws.com-1574382155-273939",
                        "DomainName": "awsexamplebucket.s3.amazonaws.com",
                        "OriginPath": "",
                        "CustomHeaders": {
                            "Quantity": 0
                        },
                        "S3OriginConfig": {
                            "OriginAccessIdentity": ""
                        }
                    }
                ]
            },
            "OriginGroups": {
                "Quantity": 0
            },
            "DefaultCacheBehavior": {
                "TargetOriginId":
"awsexamplebucket.s3.amazonaws.com-1574382155-273939",
                "ForwardedValues": {
                    "QueryString": false,
                    "Cookies": {
                        "Forward": "none"
```

```
                    },
                    "Headers": {
                        "Quantity": 0
                    },
                    "QueryStringCacheKeys": {
                        "Quantity": 0
                    }
                },
                "TrustedSigners": {
                    "Enabled": false,
                    "Quantity": 0
                },
                "ViewerProtocolPolicy": "allow-all",
                "MinTTL": 0,
                "AllowedMethods": {
                    "Quantity": 2,
                    "Items": [
                        "HEAD",
                        "GET"
                    ],
                    "CachedMethods": {
                        "Quantity": 2,
                        "Items": [
                            "HEAD",
                            "GET"
                        ]
                    }
                },
                "SmoothStreaming": false,
                "DefaultTTL": 86400,
                "MaxTTL": 31536000,
                "Compress": false,
                "LambdaFunctionAssociations": {
                    "Quantity": 0
                },
                "FieldLevelEncryptionId": ""
            },
            "CacheBehaviors": {
                "Quantity": 0
            },
            "CustomErrorResponses": {
                "Quantity": 0
            },
            "Comment": "",
```

```
            "Logging": {
                "Enabled": false,
                "IncludeCookies": false,
                "Bucket": "",
                "Prefix": ""
            },
            "PriceClass": "PriceClass_All",
            "Enabled": false,
            "ViewerCertificate": {
                "CloudFrontDefaultCertificate": true,
                "MinimumProtocolVersion": "TLSv1",
                "CertificateSource": "cloudfront"
            },
            "Restrictions": {
                "GeoRestriction": {
                    "RestrictionType": "none",
                    "Quantity": 0
                }
            },
            "WebACLId": "",
            "HttpVersion": "http2",
            "IsIPV6Enabled": true
        }
    }
}
```

- 자세한 API 내용은 명령 참조UpdateDistribution의 섹션을 참조하세요. AWS CLI

### update-field-level-encryption-config

다음 코드 예시에서는 update-field-level-encryption-config을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudFront 필드 수준 암호화 구성을 업데이트하려면

다음 예제에서는 JSON 파일에 파라미터를 C3KM2WVD605UAY 제공하여 필드 수준 암호화 구성의 Comment 필드를 ID로 업데이트합니다.

필드 수준 암호화 구성을 업데이트하려면 구성의 ID 및 가 있어야 합니다ETag. ID는 create-field-level-encryption-config 및 list-field-level-encryption-config 명령의 출력으로 반환됩니다. 를 가져오

려면 get-field-level-encryption 또는 get-field-level-encryption-config 명령을 ETag사용합니다. --if-match 옵션을 사용하여 구성의 를 제공합니다ETag.

```
aws cloudfront update-field-level-encryption-config \
    --id C3KM2WVD605UAY \
    --if-match E2P4Z4VU7TY5SG \
    --field-level-encryption-config file://fle-config.json
```

파일은 현재 디렉터리의 JSON 문서fle-config.json로, 다음을 포함합니다.

```
{
    "CallerReference": "cli-example",
    "Comment": "Updated example FLE configuration",
    "QueryArgProfileConfig": {
        "ForwardWhenQueryArgProfileIsUnknown": true,
        "QueryArgProfiles": {
            "Quantity": 0
        }
    },
    "ContentTypeProfileConfig": {
        "ForwardWhenContentTypeIsUnknown": true,
        "ContentTypeProfiles": {
            "Quantity": 1,
            "Items": [
                {
                    "Format": "URLEncoded",
                    "ProfileId": "P280MFCLSYOCVU",
                    "ContentType": "application/x-www-form-urlencoded"
                }
            ]
        }
    }
}
```

출력:

```
{
    "ETag": "E26M4BIAV81ZF6",
    "FieldLevelEncryption": {
        "Id": "C3KM2WVD605UAY",
        "LastModifiedTime": "2019-12-10T22:26:26.170Z",
        "FieldLevelEncryptionConfig": {
```

```
            "CallerReference": "cli-example",
            "Comment": "Updated example FLE configuration",
            "QueryArgProfileConfig": {
                "ForwardWhenQueryArgProfileIsUnknown": true,
                "QueryArgProfiles": {
                    "Quantity": 0,
                    "Items": []
                }
            },
            "ContentTypeProfileConfig": {
                "ForwardWhenContentTypeIsUnknown": true,
                "ContentTypeProfiles": {
                    "Quantity": 1,
                    "Items": [
                        {
                            "Format": "URLEncoded",
                            "ProfileId": "P280MFCLSYOCVU",
                            "ContentType": "application/x-www-form-urlencoded"
                        }
                    ]
                }
            }
        }
    }
}
```

- 자세한 API 내용은 명령 참조[UpdateFieldLevelEncryptionConfig](#)의 섹션을 참조하세요. AWS CLI

### update-field-level-encryption-profile

다음 코드 예시에서는 `update-field-level-encryption-profile`을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudFront 필드 수준 암호화 프로필을 업데이트하려면

다음 예제에서는 필드 수준 암호화 프로파일을 ID 로 업데이트합니다`PPK0UOSIF5WSV`. 이 예제에서는 프로파일의 `Name` 및 를 업데이트`Comment`하고 JSON 파일에 파라미터를 제공하여 두 번째 `FieldPatterns` 항목을 추가합니다.

필드 수준 암호화 프로파일을 업데이트하려면 프로파일의 ID 및 가 있어야 합니다`ETag`. ID는 create-field-level-encryption-profile 및 list-field-level-encryption-profiles 명령의 출력에 반환됩니다.

를 가져오려면 get-field-level-encryption-profile 또는 get-field-level-encryption-profile-config 명령을 ETag사용합니다. `--if-match` 옵션을 사용하여 프로필의 를 제공합니다ETag.

```
aws cloudfront update-field-level-encryption-profile \
    --id PPK0UOSIF5WSV \
    --if-match E1QQG65FS2L2GC \
    --field-level-encryption-profile-config file://fle-profile-config.json
```

파일은 현재 디렉터리의 JSON 문서fle-profile-config.json로, 다음을 포함합니다.

```
{
    "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"
                    ]
                }
            }
        ]
    }
}
```

출력:

```
{
    "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"
                            ]
                        }
                    }
                ]
            }
        }
    }
}
```

- 자세한 API 내용은 명령 참조UpdateFieldLevelEncryptionProfile의 섹션을 참조하세요. AWS CLI

## 를 사용한 Amazon CloudSearch 예제 AWS CLI

다음 코드 예제에서는 Amazon 에서 를 사용하여 작업을 수행하고 일반적인 시나리오 AWS Command Line Interface 를 구현하는 방법을 보여줍니다 CloudSearch.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `upload-documents`

다음 코드 예시에서는 `upload-documents`을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 `upload-documents` 명령은 Amazon CloudSearch 도메인에 JSON 문서 배치를 업로드합니다.

```
aws cloudsearchdomain upload-documents --endpoint-url https://doc-my-domain.us-west-1.cloudsearch.amazonaws.com --content-type application/json --documents document-batch.json
```

출력:

```
{
  "status": "success",
  "adds": 5000,
  "deletes": 0
}
```

- 자세한 API 내용은 명령 참조UploadDocuments의 섹션을 참조하세요. AWS CLI

## CloudTrail 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 CloudTrail.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

## 작업

### add-tags

다음 코드 예시에서는 add-tags을 사용하는 방법을 보여 줍니다.

AWS CLI

추적에 태그를 추가하려면

다음 add-tags 명령은 에 대한 태그를 추가합니다Trail1.

```
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
```

- 자세한 API 내용은 명령 참조AddTags의 섹션을 참조하세요. AWS CLI

### create-subscription

다음 코드 예시에서는 create-subscription을 사용하는 방법을 보여 줍니다.

AWS CLI

추적에 대한 AWS 리소스를 생성하고 구성하려면

다음 create-subscription 명령은 에 대한 새 S3 버킷 및 SNS 주제를 생성합니다Trail1.

```
aws cloudtrail create-subscription --name Trail1 --s3-new-bucket my-bucket --sns-
new-topic my-topic
```

출력:

```
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
```

- 자세한 API 내용은 명령 참조CreateSubscription의 섹션을 참조하세요. AWS CLI

### `create-trail`

다음 코드 예시에서는 `create-trail`을 사용하는 방법을 보여 줍니다.

AWS CLI

추적을 생성하려면

다음 `create-trail` 명령은 라는 다중 리전 추적을 생성하고 S3 버킷을 `Trail1` 지정합니다.

```
aws cloudtrail create-trail --name Trail1 --s3-bucket-name my-bucket --is-multi-
region-trail
```

출력:

```
{
  "IncludeGlobalServiceEvents": true,
  "Name": "Trail1",
  "TrailARN": "arn:aws:cloudtrail:us-west-2:123456789012:trail/Trail1",
  "LogFileValidationEnabled": false,
  "IsMultiRegionTrail": true,
  "S3BucketName": "my-bucket"
}
```

- 자세한 API 내용은 명령 참조CreateTrail의 섹션을 참조하세요. AWS CLI

### delete-trail

다음 코드 예시에서는 delete-trail을 사용하는 방법을 보여 줍니다.

AWS CLI

추적을 삭제하려면

다음 delete-trail 명령은 라는 추적을 삭제합니다Trail1.

```
aws cloudtrail delete-trail --name Trail1
```

- 자세한 API 내용은 명령 참조DeleteTrail의 섹션을 참조하세요. AWS CLI

### describe-trails

다음 코드 예시에서는 describe-trails을 사용하는 방법을 보여 줍니다.

AWS CLI

추적을 설명하려면

다음 describe-trails 명령은 Trail1 및 에 대한 설정을 반환합니다Trail2.

```
aws cloudtrail describe-trails --trail-name-list Trail1 Trail2
```

출력:

```
{
 "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeTrails의 섹션을 참조하세요. AWS CLI

### get-event-selectors

다음 코드 예시에서는 get-event-selectors을 사용하는 방법을 보여 줍니다.

AWS CLI

추적에 대한 이벤트 선택기 설정을 보려면

다음 get-event-selectors 명령은 에 대한 설정을 반환합니다Trail1.

```
aws cloudtrail get-event-selectors --trail-name Trail1
```

출력:

```
{
    "EventSelectors": [
        {
            "IncludeManagementEvents": true,
            "DataResources": [],
            "ReadWriteType": "All"
        }
    ],
    "TrailARN": "arn:aws:cloudtrail:us-east-1:123456789012:trail/Trail1"
```

```
}
```

- 자세한 API 내용은 명령 참조GetEventSelectors의 섹션을 참조하세요. AWS CLI

### `get-trail-status`

다음 코드 예시에서는 `get-trail-status`을 사용하는 방법을 보여 줍니다.

AWS CLI

추적 상태를 가져오려면

다음 `get-trail-status` 명령은 에 대한 전송 및 로깅 세부 정보를 반환합니다`Trail1`.

```
aws cloudtrail get-trail-status --name Trail1
```

출력:

```
{
  "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"
}
```

- 자세한 API 내용은 명령 참조GetTrailStatus의 섹션을 참조하세요. AWS CLI

### `list-public-keys`

다음 코드 예시에서는 `list-public-keys`을 사용하는 방법을 보여 줍니다.

AWS CLI

추적의 모든 퍼블릭 키를 나열하려면

다음 list-public-keys 명령은 지정된 시간 범위 내에서 다이제스트 파일에 서명하는 데 프라이빗 키가 사용된 모든 퍼블릭 키를 반환합니다.

```
aws cloudtrail list-public-keys --start-time 2016-01-01T20:30:00.000Z
```

출력:

```
{
  "PublicKeyList": [
      {
         "ValidityStartTime": 1453076702.0,
         "ValidityEndTime": 1455668702.0,
         "Value": "MIIBCgKCAQEAlSS3cl92HDycr/MTj0moOhas8habjrraXw+KzlWF0axSI2tcF
+3iJ9BKQAVSKxGwxwu3m0wG3J
+kUl1xboEcEPHYoIYMbgfSw7KGnuDKwkLzsQWhUJ0cIbOHASox1vv/5fNXkrHhGbDCHeVXm804c83nvHUEFYThr1PfyP
+4WGDk+BGH5m9iuiAKkipEHWmUl8/P7XpfpWQuk4h8g3pXZOrNXr08lbh4d39svj7UqdhvOXoBISp9t/
EXYuePGEtBdrKD9Dz+VHwyUPtBQvYr9BnkF88qBnaPNhS44rzwIDAQAB",
         "Fingerprint": "7f3f401420072e50a65a141430817ab3"
      }
  ]
}
```

- 자세한 API 내용은 명령 참조ListPublicKeys의 섹션을 참조하세요. AWS CLI

### list-tags

다음 코드 예시에서는 list-tags을 사용하는 방법을 보여 줍니다.

AWS CLI

추적의 태그를 나열하려면

다음 list-tags 명령은 Trail1 및 에 대한 태그를 나열합니다Trail2.

```
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
```

출력:

```
{
 "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"
            }
        ]
    }
  ]
}
```

- 자세한 API 내용은 명령 참조ListTags의 섹션을 참조하세요. AWS CLI

### `lookup-events`

다음 코드 예시에서는 `lookup-events`을 사용하는 방법을 보여 줍니다.

AWS CLI

추적에 대한 이벤트를 조회하려면

다음 `lookup-events` 명령은 속성 별로 API 활동 이벤트를 검색합니다`EventName`.

```
aws cloudtrail lookup-events --lookup-
attributes AttributeKey=EventName,AttributeValue=ConsoleLogin
```

출력:

```
{
```

```
  "Events": [
     {
         "EventId": "654ccbc0-ba0d-486a-9076-dbf7274677a7",
         "Username": "my-session-name",
         "EventTime": "2021-11-18T09:41:02-08:00",
         "CloudTrailEvent": "{\"eventVersion\":\"1.02\",\"userIdentity\":{\"type\":
\"AssumedRole\",\"principalId\":\"AROAJIKPFTA72SWU4L7T4:my-session-name\",\"arn\":
\"arn:aws:sts::123456789012:assumed-role/my-role/my-session-name\",\"accountId\":
\"123456789012\",\"sessionContext\":{\"attributes\":{\"mfaAuthenticated\":\"false
\",\"creationDate\":\"2016-01-26T21:42:12Z\"},\"sessionIssuer\":{\"type\":\"Role\",
\"principalId\":\"AROAJIKPFTA72SWU4L7T4\",\"arn\":\"arn:aws:iam::123456789012:role/
my-role\",\"accountId\":\"123456789012\",\"userName\":\"my-role\"}}},\"eventTime
\":\"2016-01-26T21:42:12Z\",\"eventSource\":\"signin.amazonaws.com\",\"eventName\":
\"ConsoleLogin\",\"awsRegion\":\"us-east-1\",\"sourceIPAddress\":\"72.21.198.70\",
\"userAgent\":\"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_5) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/47.0.2526.111 Safari/537.36\",\"requestParameters
\":null,\"responseElements\":{\"ConsoleLogin\":\"Success\"},\"additionalEventData\":
{\"MobileVersion\":\"No\",\"MFAUsed\":\"No\"},\"eventID\":\"654ccbc0-ba0d-486a-9076-
dbf7274677a7\",\"eventType\":\"AwsConsoleSignIn\",\"recipientAccountId\":
\"123456789012\"}",
         "EventName": "ConsoleLogin",
         "Resources": []
     }
  ]
}
```

- 자세한 API 내용은 명령 참조LookupEvents의 섹션을 참조하세요. AWS CLI

### put-event-selectors

다음 코드 예시에서는 put-event-selectors을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 고급 이벤트 선택기를 사용하여 관리 이벤트 및 데이터 이벤트를 로깅하도록 추적 구성

추적의 모든 조건 및 선택기에 대해 최대 500개의 값까지 고급 이벤트 선택기와 고급 이벤트 선택기의 조건을 추가할 수 있습니다. 고급 이벤트 선택기를 사용하여 사용 가능한 모든 데이터 이벤트 유형을 기록할 수 있습니다. 고급 이벤트 선택기 또는 기본 이벤트 선택기 중 하나를 사용할 수 있습니다. 추적에 고급 이벤트 선택기를 적용하면 기존의 기본 이벤트 선택기를 모두 덮어씁니다.

다음 예제에서는 모든 관리 이벤트를 로깅myTrail하고, S3 PutObject 버킷 하나를 제외한 모든 S3 및 DeleteObject API 호출을 로그하고, 라는 Lambda 함수에 대한 데이터 API 호출을 로그하

고myFunction, 라는 SNS 주제에 대한 API 호출을 로그 게시하기 위해 라는 추적에 대한 고급 이벤트 선택기를 생성합니다myTopic.

```
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"] }]}]'
```

출력:

```
{
    "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"
                    ]
                }
            ]
        }
    ]
}
```

자세한 내용은 AWS CloudTrail 사용 설명서의 고급 이벤트 선택기를 사용하여 이벤트 로깅을 참조하세요.

예제 2: 추적에 대한 이벤트 선택기가 모든 관리 이벤트 및 데이터 이벤트를 기록하도록 구성

최대 5개의 이벤트 선택기와 최대 250개의 데이터 리소스를 추적 대상으로 구성할 수 있습니다. 이벤트 선택기는 기본 이벤트 선택기라고도 합니다. 이벤트 선택기를 사용하여 S3 객체, Lambda 함

수 및 DynnamoDB 테이블에 대한 관리 이벤트 및 데이터 이벤트를 로깅할 수 있습니다. 다른 리소스 유형에 대한 데이터 이벤트를 로깅하려면 고급 이벤트 선택기를 사용해야 합니다.

다음 예제에서는 모든 관리 이벤트, 두 개의 Amazon S3 버킷/접두사 조합에 대한 데이터 이벤트, 라는 단일 AWS Lambda 함수에 대한 데이터 이벤트를 `TrailName` 포함하도록 라는 이름의 추적에 대한 이벤트 선택기를 생성합니다`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"]}]}]'
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS CloudTrail 사용 설명서의 기본 이벤트 선택기를 사용하여 이벤트 로깅을 참조하세요.

예제 3: 추적에 대한 이벤트 선택기가 관리 이벤트, S3 객체의 모든 S3 데이터 이벤트 및 계정의 함수에 대한 모든 Lambda 데이터 이벤트를 로깅하도록 구성

다음 예제에서는 모든 관리 이벤트와 AWS 계정의 모든 Amazon S3 버킷 및 AWS Lambda 함수에 대한 모든 데이터 이벤트를 `TrailName2` 포함하는 라는 추적에 대한 이벤트 선택기를 생성합니다.

```
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"]}]}]'
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS CloudTrail 사용 설명서의 기본 이벤트 선택기를 사용하여 이벤트 로깅을 참조하세요.

- 자세한 API 내용은 명령 참조PutEventSelectors의 섹션을 참조하세요. AWS CLI

### remove-tags

다음 코드 예시에서는 remove-tags을 사용하는 방법을 보여 줍니다.

AWS CLI

추적의 태그를 제거하려면

다음 remove-tags 명령은 에 대해 지정된 태그를 제거합니다Trail1.

```
aws cloudtrail remove-tags --resource-id arn:aws:cloudtrail:us-
east-1:123456789012:trail/Trail1 --tags-list Key=name Key=location
```

- 자세한 API 내용은 명령 참조RemoveTags의 섹션을 참조하세요. AWS CLI

### start-logging

다음 코드 예시에서는 start-logging을 사용하는 방법을 보여 줍니다.

AWS CLI

추적에 대한 로깅을 시작하려면

다음 start-logging 명령은 에 대한 로깅을 활성화합니다Trail1.

```
aws cloudtrail start-logging --name Trail1
```

- 자세한 API 내용은 명령 참조StartLogging의 섹션을 참조하세요. AWS CLI

### stop-logging

다음 코드 예시에서는 stop-logging을 사용하는 방법을 보여 줍니다.

AWS CLI

추적 로깅을 중지하려면

다음 stop-logging 명령은 에 대한 로깅을 끕니다Trail1.

```
aws cloudtrail stop-logging --name Trail1
```

- 자세한 API 내용은 명령 참조StopLogging의 섹션을 참조하세요. AWS CLI

### update-subscription

다음 코드 예시에서는 update-subscription을 사용하는 방법을 보여 줍니다.

AWS CLI

추적에 대한 구성 설정을 업데이트하려면

다음 update-subscription 명령은 추적을 업데이트하여 새 S3 버킷 및 SNS 주제를 지정합니다.

```
aws cloudtrail update-subscription --name Trail1 --s3-new-bucket my-bucket-new --
sns-new-topic my-topic-new
```

출력:

```
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"
}
}
```

- 자세한 API 내용은 명령 참조UpdateSubscription의 섹션을 참조하세요. AWS CLI

### update-trail

다음 코드 예시에서는 update-trail을 사용하는 방법을 보여 줍니다.

AWS CLI

추적을 업데이트하려면

다음 update-trail 명령은 로그 전송에 기존 버킷을 사용하도록 추적을 업데이트합니다.

```
aws cloudtrail update-trail --name Trail1 --s3-bucket-name my-bucket
```

출력:

```
{
  "IncludeGlobalServiceEvents": true,
  "Name": "Trail1",
  "TrailARN": "arn:aws:cloudtrail:us-west-2:123456789012:trail/Trail1",
  "LogFileValidationEnabled": false,
  "IsMultiRegionTrail": true,
  "S3BucketName": "my-bucket"
}
```

- 자세한 API 내용은 명령 참조UpdateTrail의 섹션을 참조하세요. AWS CLI

### validate-logs

다음 코드 예시에서는 validate-logs을 사용하는 방법을 보여 줍니다.

AWS CLI

로그 파일을 검증하려면

다음 validate-logs 명령은 에 대한 로그를 검증합니다Trail1.

```
aws cloudtrail validate-logs --trail-arn arn:aws:cloudtrail:us-
east-1:123456789012:trail/Trail1 --start-time 20160129T19:00:00Z
```

출력:

```
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
```

- 자세한 API 내용은 명령 참조ValidateLogs의 섹션을 참조하세요. AWS CLI

## CloudWatch 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 CloudWatch.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `delete-alarms`

다음 코드 예시에서는 `delete-alarms`을 사용하는 방법을 보여 줍니다.

AWS CLI

경보를 삭제하는 방법

다음 예제에서는 delete-alarms 명령을 사용하여 'myalarm'이라는 Amazon CloudWatch 경보를 삭제합니다.

```
aws cloudwatch delete-alarms --alarm-names myalarm
```

출력:

```
This command returns to the prompt if successful.
```

- 자세한 API 내용은 명령 참조DeleteAlarms의 섹션을 참조하세요. AWS CLI

### describe-alarm-history

다음 코드 예시에서는 describe-alarm-history을 사용하는 방법을 보여 줍니다.

AWS CLI

경보에 대한 기록을 검색하는 방법

다음 예제에서는 describe-alarm-history 명령을 사용하여 “myalarm”이라는 Amazon CloudWatch 경보의 기록을 검색합니다.

```
aws cloudwatch describe-alarm-history --alarm-name "myalarm" --history-item-
type StateUpdate
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeAlarmHistory의 섹션을 참조하세요. AWS CLI

### `describe-alarms-for-metric`

다음 코드 예시에서는 `describe-alarms-for-metric`을 사용하는 방법을 보여 줍니다.

AWS CLI

지표와 관련된 경보에 대한 정보를 표시하는 방법

다음 예제에서는 `describe-alarms-for-metric` 명령을 사용하여 Amazon 지표 및 IDEC2CPUUtilization가 i-0c986c72인 인스턴스와 연결된 경보에 대한 정보를 표시합니다.

```
aws cloudwatch describe-alarms-for-metric --metric-name CPUUtilization --
namespace AWS/EC2 --dimensions Name=InstanceId,Value=i-0c986c72
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeAlarmsForMetric의 섹션을 참조하세요. AWS CLI

### `describe-alarms`

다음 코드 예시에서는 `describe-alarms`을 사용하는 방법을 보여 줍니다.

AWS CLI

경보에 대한 정보를 나열하는 방법

다음 예제에서는 `describe-alarms` 명령을 사용하여 'myalarm'이라는 경보에 대한 정보를 제공합니다.

```
aws cloudwatch describe-alarms --alarm-names "myalarm"
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeAlarms의 섹션을 참조하세요. AWS CLI

### disable-alarm-actions

다음 코드 예시에서는 disable-alarm-actions을 사용하는 방법을 보여 줍니다.

AWS CLI

경보에 대한 작업을 비활성화하는 방법

다음 예제에서는 disable-alarm-actions 명령을 사용하여 myalarm이라는 경보에 대한 모든 작업을 비활성화합니다.

```
aws cloudwatch disable-alarm-actions --alarm-names myalarm
```

이 명령은 성공하면 프롬프트로 돌아갑니다.

- 자세한 API 내용은 명령 참조DisableAlarmActions의 섹션을 참조하세요. AWS CLI

### enable-alarm-actions

다음 코드 예시에서는 enable-alarm-actions을 사용하는 방법을 보여 줍니다.

AWS CLI

경보에 대한 모든 작업을 활성화하는 방법

다음 예제에서는 enable-alarm-actions 명령을 사용하여 myalarm이라는 경보에 대한 모든 작업을 활성화합니다.

```
aws cloudwatch enable-alarm-actions --alarm-names myalarm
```

이 명령은 성공하면 프롬프트로 돌아갑니다.

- 자세한 API 내용은 명령 참조EnableAlarmActions의 섹션을 참조하세요. AWS CLI

### get-metric-statistics

다음 코드 예시에서는 get-metric-statistics을 사용하는 방법을 보여 줍니다.

AWS CLI

EC2 인스턴스당 CPU 사용률을 가져오려면

다음 예제에서는 `get-metric-statistics` 명령을 사용하여 ID가 i-abcdef인 EC2 인스턴스의 CPU 사용률을 가져옵니다.

```
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
```

출력:

```
{
    "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"
}
```

여러 측정기준을 지정하는 방법

다음 예제는 여러 측정기준을 지정하는 방법을 보여줍니다. 각 측정기준은 이름과 값 사이에 쉼표가 있는 이름/값 페어로 지정됩니다. 여러 측정기준은 공백으로 구분됩니다. 단일 지표에 여러 개의 측정기준이 포함된 경우에는 정의된 모든 측정기준에 대해 값을 지정해야 합니다.

`get-metric-statistics` 명령을 사용하는 자세한 예는 Amazon CloudWatch 개발자 안내서의 지표에 대한 통계 가져오기를 참조하세요.

```
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
```

- 자세한 API 내용은 명령 참조GetMetricStatistics의 섹션을 참조하세요. AWS CLI

### `list-metrics`

다음 코드 예시에서는 `list-metrics`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon의 지표를 나열하려면 SNS

다음 `list-metrics` 예제에서는 Amazon 에 대한 지표를 보여줍니다SNS.

```
aws cloudwatch list-metrics \
    --namespace "AWS/SNS"
```

출력:

```
{
    "Metrics": [
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "NotifyMe"
                }
            ],
            "MetricName": "PublishSize"
```

```
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "CFO"
                }
            ],
            "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": "CFO"
                }
            ],
            "MetricName": "NumberOfMessagesPublished"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "CFO"
                }
            ],
            "MetricName": "NumberOfNotificationsDelivered"
        },
        {
            "Namespace": "AWS/SNS",
            "Dimensions": [
                {
                    "Name": "TopicName",
                    "Value": "CFO"
                }
            ],
            "MetricName": "NumberOfNotificationsFailed"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListMetrics의 섹션을 참조하세요. AWS CLI

### put-metric-alarm

다음 코드 예시에서는 put-metric-alarm을 사용하는 방법을 보여 줍니다.

AWS CLI

CPU 사용률이 70%를 초과할 때 Amazon Simple Notification Service 이메일 메시지를 보내려면

다음 예제에서는 CPU 사용률이 70%를 초과할 때 put-metric-alarm 명령을 사용하여 Amazon Simple Notification Service 이메일 메시지를 보냅니다.

```
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
```

이 명령은 성공하면 프롬프트로 돌아갑니다. 같은 이름의 경보가 이미 있는 경우 새 경보가 해당 경보를 덮어씁니다.

여러 측정기준을 지정하는 방법

다음 예제는 여러 측정기준을 지정하는 방법을 보여줍니다. 각 측정기준은 이름과 값 사이에 쉼표가 있는 이름/값 페어로 지정됩니다. 여러 측정기준은 공백으로 구분됩니다.

```
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
```

- 자세한 API 내용은 명령 참조PutMetricAlarm의 섹션을 참조하세요. AWS CLI

### put-metric-data

다음 코드 예시에서는 put-metric-data을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 지표를 Amazon에 게시하려면 CloudWatch

다음 예제에서는 put-metric-data 명령을 사용하여 사용자 지정 지표를 Amazon 에 게시합니다 CloudWatch.

```
aws cloudwatch put-metric-data --namespace "Usage Metrics" --metric-data file://
metric.json
```

지표 자체의 값은 JSON 파일 에 저장됩니다metric.json.

해당 파일의 내용은 다음과 같습니다.

```
[
  {
    "MetricName": "New Posts",
    "Timestamp": "Wednesday, June 12, 2013 8:28:20 PM",
    "Value": 0.50,
    "Unit": "Count"
  }
]
```

자세한 내용은 Amazon CloudWatch 개발자 안내서의 사용자 지정 지표 게시를 참조하세요.

여러 측정기준을 지정하는 방법

다음 예제는 여러 측정기준을 지정하는 방법을 보여줍니다. 각 측정기준은 이름=값 페어로 지정됩니다. 여러 측정기준은 쉼표로 구분됩니다.

```
aws cloudwatch put-metric-data --metric-name Buffers --
namespace MyNameSpace --unit Bytes --value 231434333 --
dimensions InstanceID=1-23456789,InstanceType=m1.small
```

- 자세한 API 내용은 명령 참조PutMetricData의 섹션을 참조하세요. AWS CLI

### set-alarm-state

다음 코드 예시에서는 set-alarm-state을 사용하는 방법을 보여 줍니다.

AWS CLI

경보 상태를 일시적으로 변경하려면

다음 예제에서는 set-alarm-state 명령을 사용하여 'myalarm'이라는 Amazon CloudWatch 경보의 상태를 일시적으로 변경하고 테스트 목적으로 ALARM 상태로 설정합니다.

```
aws cloudwatch set-alarm-state --alarm-name "myalarm" --state-value ALARM --state-
reason "testing purposes"
```

이 명령은 성공하면 프롬프트로 돌아갑니다.

- 자세한 API 내용은 명령 참조SetAlarmState의 섹션을 참조하세요. AWS CLI

# CloudWatch 를 사용하여 예제를 로깅합니다. AWS CLI

다음 코드 예제에서는 CloudWatch 로그 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `create-log-group`

다음 코드 예시에서는 `create-log-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 이름이 `my-logs`인 로그 그룹을 생성합니다.

```
aws logs create-log-group --log-group-name my-logs
```

- 자세한 API 내용은 명령 참조CreateLogGroup의 섹션을 참조하세요. AWS CLI

### `create-log-stream`

다음 코드 예시에서는 `create-log-stream`을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 `my-logs` 로그 그룹에서 이름이 `20150601`인 로그 스트림을 생성합니다.

```
aws logs create-log-stream --log-group-name my-logs --log-stream-name 20150601
```

- 자세한 API 내용은 명령 참조CreateLogStream의 섹션을 참조하세요. AWS CLI

### delete-log-group

다음 코드 예시에서는 delete-log-group을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 이름이 my-logs인 로그 그룹을 삭제합니다.

```
aws logs delete-log-group --log-group-name my-logs
```

- 자세한 API 내용은 명령 참조DeleteLogGroup의 섹션을 참조하세요. AWS CLI

### delete-log-stream

다음 코드 예시에서는 delete-log-stream을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 라는 로그 그룹에서 라는 로그 스트림20150531을 삭제합니다my-logs.

```
aws logs delete-log-stream --log-group-name my-logs --log-stream-name 20150531
```

- 자세한 API 내용은 명령 참조DeleteLogStream의 섹션을 참조하세요. AWS CLI

### delete-retention-policy

다음 코드 예시에서는 delete-retention-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 라는 로그 그룹에 이전에 적용된 보존 정책을 제거합니다my-logs.

```
aws logs delete-retention-policy --log-group-name my-logs
```

- 자세한 API 내용은 명령 참조DeleteRetentionPolicy의 섹션을 참조하세요. AWS CLI

### describe-log-groups

다음 코드 예시에서는 describe-log-groups을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 이름이 my-logs인 로그 그룹을 설명합니다.

```
aws logs describe-log-groups --log-group-name-prefix my-logs
```

출력:

```
{
    "logGroups": [
        {
            "storedBytes": 0,
            "metricFilterCount": 0,
            "creationTime": 1433189500783,
            "logGroupName": "my-logs",
            "retentionInDays": 5,
            "arn": "arn:aws:logs:us-west-2:0123456789012:log-group:my-logs:*"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeLogGroups의 섹션을 참조하세요. AWS CLI

### describe-log-streams

다음 코드 예시에서는 describe-log-streams을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 로그 그룹 2015의 접두사로 시작하는 모든 로그 스트림을 보여줍니다my-logs.

```
aws logs describe-log-streams --log-group-name my-logs --log-stream-name-prefix 2015
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeLogStreams의 섹션을 참조하세요. AWS CLI

### get-log-events

다음 코드 예시에서는 get-log-events을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 로그 그룹 20150601에 이름이 지정된 로그 스트림에서 로그 이벤트를 검색합니다my-logs.

```
aws logs get-log-events --log-group-name my-logs --log-stream-name 20150601
```

출력:

```
{
    "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"
}
```

- 자세한 API 내용은 명령 참조GetLogEvents의 섹션을 참조하세요. AWS CLI

### put-log-events

다음 코드 예시에서는 put-log-events을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 로그 그룹 20150601에 이름이 지정된 로그 스트림에 로그 이벤트를 넣습니다my-logs.

```
aws logs put-log-events --log-group-name my-logs --log-stream-name 20150601 --log-events file://events
```

출력:

```
{
    "nextSequenceToken": "49542672486831074009579604567656788214806863282469607346"
}
```

위 예제는 현재 디렉터리events에 이름이 지정된 파일에서 이벤트 JSON 배열을 읽습니다.

```
[
  {
    "timestamp": 1433190184356,
    "message": "Example Event 1"
  },
  {
    "timestamp": 1433190184358,
    "message": "Example Event 2"
  },
  {
```

```
    "timestamp": 1433190184360,
    "message": "Example Event 3"
  }
]
```

후속 호출마다 이전 호출에서 제공하는 다음 시퀀스 토큰을 시퀀스 토큰 옵션으로 지정해야 합니다.

```
aws logs put-log-events --log-group-name my-logs --log-
stream-name 20150601 --log-events file://events2 --sequence-
token "49542672486831074009579604567656788214806863282469607346"
```

출력:

```
{
    "nextSequenceToken": "49542672486831074009579604567900991230369019956308219826"
}
```

- 자세한 API 내용은 명령 참조PutLogEvents의 섹션을 참조하세요. AWS CLI

### put-retention-policy

다음 코드 예시에서는 put-retention-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 라는 로그 그룹에 5일 보존 정책을 추가합니다my-logs.

```
aws logs put-retention-policy --log-group-name my-logs --retention-in-days 5
```

- 자세한 API 내용은 명령 참조PutRetentionPolicy의 섹션을 참조하세요. AWS CLI

## CloudWatch 를 사용한 네트워크 모니터링 예제 AWS CLI

다음 코드 예제에서는 CloudWatch 네트워크 모니터링과 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `create-monitor`

다음 코드 예시에서는 `create-monitor`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 집계 기간이 있는 네트워크 모니터를 생성하려면

다음 `create-monitor` 예제에서는 `30` 초로 `aggregationPeriod` 설정된 의 이름이 `Example_NetworkMonitor` 지정된 모니터를 생성합니다. 모니터와 연결된 프로브가 없기 `INACTIVE` 때문에 모니터`state`의 이니셜이 됩니다. 프로브가 추가될 때`ACTIVE`만 상태가 로 변경됩니다. update-monitor 또는 create-probe 명령을 사용하여 이 모니터에 프로브를 추가할 수 있습니다.

```
aws networkmonitor create-monitor \
     --monitor-name Example_NetworkMonitor \
     --aggregation-period 30
```

출력:

```
{
    "monitorArn": "arn:aws:networkmonitor:region:111122223333:monitor/
Example_NetworkMonitor",
    "monitorName": "Example_NetworkMonitor",
    "state": "INACTIVE",
    "aggregationPeriod": 30,
    "tags": {}
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon CloudWatch Network Monitor 작동 방식을 참조하세요.  CloudWatch

예제 2: 를 사용하여 프로브로 네트워크 모니터를 생성`TCP`하려면 및 에 태그도 포함

다음 create-monitor 예제에서는 라는 모니터를 생성합니다Example_NetworkMonitor. 또한 명령은 ICMP 프로토콜을 사용하고 태그를 포함하는 하나의 프로브를 생성합니다. 요청에 전달aggregationPeriod되지 않으므로 60 초가 기본값으로 설정됩니다. 프로브가 있는 모니터state의 는 모니터가 가 될 PENDING 때까지 유지됩니다ACTIVE. 이 작업은 몇 분 정도 걸릴 수 있으며, 이 시점에서 state 가 로 변경ACTIVE되고 CloudWatch 지표 보기를 시작할 수 있습니다.

```
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
```

출력:

```
{
    "monitorArn": "arn:aws:networkmonitor:region111122223333:monitor/
Example_NetworkMonitor",
    "monitorName": "Example_NetworkMonitor",
    "state": "PENDING",
    "aggregationPeriod": 60,
    "tags": {
        "Monitor": "Monitor1"
    }
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon CloudWatch Network Monitor 작동 방식을 참조하세요. CloudWatch

예제 3: 를 사용하여 프로브로 네트워크 모니터를 생성ICMP하려면 및 에 태그도 포함

다음 create-monitor 예제에서는 aggregationPeriod 의 30 이름이 초Example_NetworkMonitor인 모니터를 생성합니다. 또한 명령은 ICMP 프로토콜을 사용하고 태그를 포함하는 하나의 프로브를 생성합니다. 요청에 전달aggregationPeriod되지 않으므로 60 초가 기본값으로 설정됩니다. 프로브가 있는 모니터state의 는 모니터가 가 될 PENDING 때까지 유지됩니다ACTIVE. 이 작업은 몇 분 정도 걸릴 수 있으며, 이 시점에서 state 가 로 변경ACTIVE되고 CloudWatch 지표 보기를 시작할 수 있습니다.

```
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
```

출력:

```
{
    "monitorArn": "arn:aws:networkmonitor:region:111122223333:monitor/
Example_NetworkMonitor",
    "monitorName": "Example_NetworkMonitor",
    "state": "PENDING",
    "aggregationPeriod": 30,
    "tags": {
        "Monitor": "Monitor1"
    }
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon CloudWatch Network Monitor 작동 방식을 참조하세요. CloudWatch

- 자세한 API 내용은 명령 참조CreateMonitor의 섹션을 참조하세요. AWS CLI

### create-probe

다음 코드 예시에서는 create-probe을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 를 TCP 사용하여 프로브를 생성하고 네트워크 모니터에 추가하는 방법

다음 create-probe 예제에서는 를 사용하는 프로브를 생성하고 라는 모니터에 프로브를 TCP protocol 추가합니다Example_NetworkMonitor. 일단 생성되면 프로브가 있는 모니터state의 는 모니터가 가 될 PENDING 때까지 유지됩니다ACTIVE. 이 작업은 몇 분 정도 걸릴 수 있으며, 이 때 상태가 로 변경ACTIVE되고 CloudWatch 지표 보기를 시작할 수 있습니다.

```
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}
```

출력:

```
{
    "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"
    }
}
```

예제 2: 를 사용하여 프로브를 사용하는 프로브를 생성하고 이를 네트워크 모니터에 ICMP 추가하려면

다음 `create-probe` 예제에서는 를 사용하는 프로브를 생성하고 라는 모니터에 프로브를 `ICMP protocol` 추가합니다`Example_NetworkMonitor`. 일단 생성되면 프로브가 있는 모니터`state`의 는 모니터가 가 될 `PENDING` 때까지 유지됩니다`ACTIVE`. 이 작업은 몇 분 정도 걸릴 수 있으며, 이 때 상태가 로 변경`ACTIVE`되고 CloudWatch 지표 보기를 시작할 수 있습니다.

```
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}
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon CloudWatch Network Monitor 작동 방식을 참조하세요. CloudWatch

- 자세한 API 내용은 명령 참조CreateProbe의 섹션을 참조하세요. AWS CLI

### delete-monitor

다음 코드 예시에서는 delete-monitor을 사용하는 방법을 보여 줍니다.

AWS CLI

모니터를 삭제하려면

다음 delete-monitor 예제에서는 이름이 인 모니터를 삭제합니다Example_NetworkMonitor.

```
aws networkmonitor delete-monitor \
    --monitor-name Example_NetworkMonitor
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon 사용 설명서의 Amazon CloudWatch Network Monitor 작동 방식을 참조하세요. CloudWatch

- 자세한 API 내용은 명령 참조DeleteMonitor의 섹션을 참조하세요. AWS CLI

### delete-probe

다음 코드 예시에서는 delete-probe을 사용하는 방법을 보여 줍니다.

AWS CLI

프로브를 삭제하려면

다음 delete-probe 예제에서는 이름이 인 네트워크 모니터probe-12345에서 ID가 인 프로브를 삭제합니다Example_NetworkMonitor.

```
aws networkmonitor delete-probe \
```

```
    --monitor-name Example_NetworkMonitor \
    --probe-id probe-12345
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon 사용 설명서의 Amazon CloudWatch Network Monitor 작동 방식을 참조하세요. CloudWatch

- 자세한 API 내용은 명령 참조DeleteProbe의 섹션을 참조하세요. AWS CLI

### get-monitor

다음 코드 예시에서는 get-monitor을 사용하는 방법을 보여 줍니다.

AWS CLI

모니터 정보를 가져오려면

다음 get-monitor 예제에서는 이름이 인 모니터에 대한 정보를 가져옵니다Example_NetworkMonitor.

```
aws networkmonitor get-monitor \
    --monitor-name Example_NetworkMonitor
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon CloudWatch Network Monitor 작동 방식을 참조하세요. CloudWatch

- 자세한 API 내용은 명령 참조GetMonitor의 섹션을 참조하세요. AWS CLI

### get-probe

다음 코드 예시에서는 get-probe을 사용하는 방법을 보여 줍니다.

AWS CLI

프로브 세부 정보를 보려면

다음 get-probe 예제에서는 라는 모니터와 연결된 가 있는 프로브에 대한 세부 정보를 반환probeIDprobe-12345합니다Example_NetworkMonitor.

```
aws networkmonitor get-probe \
    --monitor-name Example_NetworkMonitor \
    --probe-id probe-12345
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon CloudWatch Network Monitor 작동 방식을 참조하세요. CloudWatch

- 자세한 API 내용은 명령 참조GetProbe의 섹션을 참조하세요. AWS CLI

### list-monitors

다음 코드 예시에서는 list-monitors을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 모든 모니터를 나열하려면(단일 모니터)

다음 `list-monitors` 예제에서는 단일 모니터의 목록만 반환합니다. 모니터의 `state`는 `ACTIVE`이고 모니터`aggregationPeriod`의 는 60초입니다.

```
aws networkmonitor list-monitors
```

출력:

```
{
    "monitors": [{
            "monitorArn": "arn:aws:networkmonitor:region:012345678910:monitor/
Example_NetworkMonitor",
            "monitorName": "Example_NetworkMonitor",
            "state": "ACTIVE",
            "aggregationPeriod": 60,
            "tags": {
                "Monitor": "Monitor1"
            }
        }
    ]
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon CloudWatch Network Monitor 작동 방식을 참조하세요.  CloudWatch

예제 2: 모든 모니터를 나열하려면(여러 모니터)

다음 `list-monitors` 예제에서는 3개의 모니터 목록을 반환합니다. 한 모니터`state`의 는 `ACTIVE` 및 CloudWatch 지표 생성입니다. 다른 두 모니터의 상태는 CloudWatch 지표를 생성`INACTIVE`하거나 생성하지 않습니다. 세 모니터 모두 60초`aggregationPeriod`의 를 사용합니다.

```
aws networkmonitor list-monitors
```

출력:

```
{
    "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": {}
        }
    ]
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon CloudWatch Network Monitor 작동 방식을 참조하세요. CloudWatch

- 자세한 API 내용은 명령 참조ListMonitors의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스의 태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 이름이 인 모니터의 태그 목록을 반환합니다`Example_NetworkMonitor`.

```
aws networkmonitor list-tags-for-resource \
    --resource-arn arn:aws:networkmonitor:region:012345678910:monitor/
Example_NetworkMonitor
```

출력:

```
{
    "tags": {
        "Environment": "Dev",
        "Application": "PetStore"
    }
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon CloudWatch Network Monitor 작동 방식을 참조하세요. CloudWatch

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### tag-resource

다음 코드 예시에서는 tag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 태그를 지정하려면

다음 tag-resource 예제에서는 Environment=Dev 및 태그Example_NetworkMonitor로 이름이 지정된 모니터에 Application=PetStore 태그를 지정합니다.

```
aws networkmonitor tag-resource \
    --resource-arn arn:aws:networkmonitor:region:012345678910:monitor/
Example_NetworkMonitor \
    --tags Environment=Dev,Application=PetStore
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon 사용 설명서의 Amazon CloudWatch Network Monitor 작동 방식을 참조하세요. CloudWatch

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### untag-resource

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스의 태그를 해제하려면

다음 untag-resource 예제에서는 키-값 페어가 인 tag-keys 파라미터를 라는 모니터와의 연결Environment Application에서 제거합니다Example_NetworkMonitor.

```
aws networkmonitor untag-resource \
    --resource-arn arn:aws:networkmonitor:region:012345678910:monitor/
Example_NetworkMonitor \
    --tag-keys Environment Application
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon 사용 설명서의 Amazon CloudWatch Network Monitor 작동 방식을 참조하세요. CloudWatch

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-monitor

다음 코드 예시에서는 update-monitor을 사용하는 방법을 보여 줍니다.

AWS CLI

모니터를 업데이트하려면

다음 update-monitor 예제에서는 모니터를 60 초aggregationPeriod에서 30 초로 변경합니다.

```
aws networkmonitor update-monitor \
    --monitor-name Example_NetworkMonitor \
    --aggregation-period 30
```

출력:

```
{
    "monitorArn": "arn:aws:networkmonitor:region:012345678910:monitor/
Example_NetworkMonitor",
```

```
    "monitorName": "Example_NetworkMonitor",
    "state": "PENDING",
    "aggregationPeriod": 30,
    "tags": {
        "Monitor": "Monitor1"
    }
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon CloudWatch Network Monitor 작동 방식을 참조하세요. CloudWatch

- 자세한 API 내용은 명령 참조UpdateMonitor의 섹션을 참조하세요. AWS CLI

### update-probe

다음 코드 예시에서는 update-probe을 사용하는 방법을 보여 줍니다.

AWS CLI

프로브를 업데이트하려면

다음 update-probe 예제에서는 프로브의 원래 destination IP 주소를 업데이트하고 도 packetSize로 업데이트합니다60.

```
aws networkmonitor update-probe \
    --monitor-name Example_NetworkMonitor \
    --probe-id probe-12345 \
    --destination 10.0.0.150 \
    --packet-size 60
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon CloudWatch Network Monitor 작동 방식을 참조하세요.  CloudWatch

- 자세한 API 내용은 명령 참조UpdateProbe의 섹션을 참조하세요. AWS CLI

## CodeArtifact 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 CodeArtifact.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `associate-external-connection`

다음 코드 예시에서는 `associate-external-connection`을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리에 외부 연결을 추가하려면

다음 `associate-external-connection` 예제에서는 npmjs.com 외부 연결을 test-repo라는 리포지토리에 추가합니다.

```
aws codeartifact associate-external-connection \
```

```
    --repository test-repo \
    --domain test-domain \
    --external-connection public:npmjs
```

출력:

```
{
    "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"
            }
        ]
    }
}
```

자세한 내용은 AWS CodeArtifact 사용 설명서의 외부 연결 추가를 참조하세요.

- 자세한 API 내용은 명령 참조AssociateExternalConnection의 섹션을 참조하세요. AWS CLI

### copy-package-versions

다음 코드 예시에서는 copy-package-versions을 사용하는 방법을 보여 줍니다.

AWS CLI

패키지 버전을 한 리포지토리에서 다른 리포지토리로 복사하려면

다음은 test-package라는 패키지의 버전 4.0.0 및 5.0.0을 my-repo에서 test-repo로 copy-package-versions 이동합니다.

```
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"]'
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS CodeArtifact 사용 설명서의 리포지토리 간에 패키지 복사를 참조하세요.

- 자세한 API 내용은 명령 참조CopyPackageVersions의 섹션을 참조하세요. AWS CLI

### create-domain

다음 코드 예시에서는 create-domain을 사용하는 방법을 보여 줍니다.

AWS CLI

도메인을 생성하려면

다음 create-domain 예제에서는 test-domain이라는 도메인을 생성합니다.

```
aws codeartifact create-domain \
    --domain test-domain
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 AWS CodeArtifact 사용 설명서의 도메인 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateDomain의 섹션을 참조하세요. AWS CLI

### `create-repository`

다음 코드 예시에서는 `create-repository`을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리를 만들려면

다음 `create-repository` 예제에서는 test-domain이라는 도메인 내에 test-repo라는 리포지토리를 생성합니다.

```
aws codeartifact create-repository \
    --domain test-domain \
    --domain-owner 111122223333 \
    --repository test-repo \
    --description "This is a test repository."
```

출력:

```
{
    "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": []
    }
}
```

자세한 내용은 AWS CodeArtifact 사용 설명서의 도메인 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateRepository의 섹션을 참조하세요. AWS CLI

### delete-domain-permissions-policy

다음 코드 예시에서는 delete-domain-permissions-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

도메인에서 권한 정책 문서를 삭제하려면

다음 delete-domain-permissions-policy 예제에서는 테스트 도메인이라는 도메인에서 권한 정책을 삭제합니다.

```
aws codeartifact delete-domain-permissions-policy \
    --domain test-domain
```

출력:

```
{
    "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"
            }
        }
    ]
}
```

자세한 내용은 AWS CodeArtifact 사용 설명서의 도메인 정책 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteDomainPermissionsPolicy의 섹션을 참조하세요. AWS CLI

### `delete-domain`

다음 코드 예시에서는 `delete-domain`을 사용하는 방법을 보여 줍니다.

AWS CLI

도메인을 삭제하려면

다음 `delete-domain` 예제에서는 라는 도메인을 삭제합니다`test-domain`.

```
aws codeartifact delete-domain \
    --domain test-domain
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 AWS CodeArtifact 사용 설명서의 도메인 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteDomain의 섹션을 참조하세요. AWS CLI

### delete-package-versions

다음 코드 예시에서는 delete-package-versions을 사용하는 방법을 보여 줍니다.

AWS CLI

패키지 버전을 삭제하려면

다음 delete-package-versions 예제에서는 test-package라는 패키지의 버전 4.0.0을 삭제합니다.

```
aws codeartifact delete-package-versions \
    --domain test-domain \
    --repo test-repo \
    --format npm \
    --package test-package \
    --versions 4.0.0
```

출력:

```
{
    "successfulVersions": {
        "4.0.0": {
            "revision": "Ciqe5/9yicvkJT13b5/LdLpCyE6fqA7poa9qp+FilPs=",
            "status": "Deleted"
        }
    },
    "failedVersions": {}
}
```

자세한 내용은 AWS CodeArtifact 사용 설명서의 패키지 버전 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeletePackageVersions의 섹션을 참조하세요. AWS CLI

### delete-repository-permissions-policy

다음 코드 예시에서는 delete-repository-permissions-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리에서 권한 정책을 삭제하려면

다음 delete-repository-permissions-policy 예제에서는 test-repo라는 리포지토리에서 권한 정책을 삭제합니다.

```
aws codeartifact delete-repository-permissions-policy \
    --domain test-domain \
    --repository test-repo
```

출력:

```
{
    "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": "*"
        }
    ]
}
```

자세한 내용은 AWS CodeArtifact 사용 설명서의 정책 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteRepositoryPermissionsPolicy의 섹션을 참조하세요. AWS CLI

### delete-repository

다음 코드 예시에서는 delete-repository을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리를 삭제하려면

다음 delete-repository 예제에서는 라는 도메인test-repo에 이름이 지정된 리포지토리를 삭제합니다test-domain.

```
aws codeartifact delete-repository \
    --domain test-domain \
    --repository test-repo
```

출력:

```
{
    "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": []
    }
}
```

자세한 내용은 AWS CodeArtifact 사용 설명서의 리포지토리 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteRepository의 섹션을 참조하세요. AWS CLI

### describe-domain

다음 코드 예시에서는 describe-domain을 사용하는 방법을 보여 줍니다.

AWS CLI

도메인에 대한 정보를 가져오려면

다음 describe-domain 예제에서는 테스트 도메인이라는 도메인의 DomainDescription 객체를 반환합니다.

```
aws codeartifact describe-domain \
    --domain test-domain
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS CodeArtifact 사용 설명서의 도메인 개요를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeDomain의 섹션을 참조하세요. AWS CLI

### describe-repository

다음 코드 예시에서는 describe-repository을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리에 대한 정보를 가져오려면

다음 describe-repository 예제에서는 test-repo라는 리포지토리의 RepositoryDescription 객체를 반환합니다.

```
aws codeartifact describe-repository \
    --domain test-domain \
    --repository test-repo
```

출력:

```
{
```

```
    "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": []
    }
}
```

자세한 내용은 AWS CodeArtifact 사용 설명서의 도메인 생성을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeRepository의 섹션을 참조하세요. AWS CLI

### `disassociate-external-connection`

다음 코드 예시에서는 `disassociate-external-connection`을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리에서 외부 연결을 제거하려면

다음 `disassociate-external-connection` 예제에서는 test-repo라는 리포지토리에서 npmjs.com 외부 연결을 제거합니다.

```
aws codeartifact disassociate-external-connection \
    --repository test-repo \
    --domain test-domain \
    --external-connection public:npmjs
```

출력:

```
{
    "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": []
    }
}
```

자세한 내용은 AWS CodeArtifact 사용 설명서의 외부 연결 제거를 참조하세요.

- 자세한 API 내용은 명령 참조DisassociateExternalConnection의 섹션을 참조하세요. AWS CLI

### `dispose-package-versions`

다음 코드 예시에서는 `dispose-package-versions`을 사용하는 방법을 보여 줍니다.

AWS CLI

패키지 버전의 자산을 삭제하고 상태를 처리됨으로 설정하려면

다음 `dispose-package-versions` 예제에서는 테스트 패키지 버전 4.0.0의 자산을 삭제하고 상태를 Disposed로 설정합니다.

```
aws codeartifact dispose-package-versions \
    --domain test-domain \
    --repo test-repo \
    --format npm \
    --package test-package \
    --versions 4.0.0
```

출력:

```
{
    "successfulVersions": {
        "4.0.0": {
            "revision": "Ciqe5/9yicvkJT13b5/LdLpCyE6fqA7poa9qp+FilPs=",
            "status": "Disposed"
        }
    },
    "failedVersions": {}
}
```

자세한 내용은 AWS CodeArtifact 사용 설명서의 에서 패키지 작업을 CodeArtifact 참조하세요.

- 자세한 API 내용은 명령 참조DisposePackageVersions의 섹션을 참조하세요. AWS CLI

### get-authorization-token

다음 코드 예시에서는 get-authorization-token을 사용하는 방법을 보여 줍니다.

AWS CLI

권한 부여 토큰을 가져오려면

다음 get-authorization-token 예제에서는 CodeArtifact 권한 부여 토큰을 검색합니다.

```
aws codeartifact get-authorization-token \
    --domain test-domain \
    --query authorizationToken \
    --output text
```

출력:

```
This command will return the authorization token. You can store the output in an
 environment variable when calling the command.
```

자세한 내용은 AWS CodeArtifact 사용 설명서의 로그인 명령 없이 pip 구성을 참조하세요.

- 자세한 API 내용은 명령 참조GetAuthorizationToken의 섹션을 참조하세요. AWS CLI

### get-domain-permissions-policy

다음 코드 예시에서는 get-domain-permissions-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

도메인에 대한 권한 정책 문서를 가져오려면

다음 get-domain-permissions-policy 예제에서는 test-domain이라는 도메인에 권한 정책을 연결합니다.

```
aws codeartifact get-domain-permissions-policy \
    --domain test-domain
```

출력:

```
{
    "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"
            }
        }
    ]
}
```

자세한 내용은 AWS CodeArtifact 사용 설명서의 도메인 정책 읽기를 참조하세요.

- 자세한 API 내용은 명령 참조GetDomainPermissionsPolicy의 섹션을 참조하세요. AWS CLI

### get-package-version-asset

다음 코드 예시에서는 get-package-version-asset을 사용하는 방법을 보여 줍니다.

AWS CLI

패키지 버전에서 자산을 가져오려면

다음 get-package-version-asset 예제에서는 test-package라는 npm 패키지의 버전 4.0.0에 대한 package.tgz 자산을 검색합니다.

```
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
```

출력:

```
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="
}
```

자세한 내용은 AWS CodeArtifact 사용 설명서의 패키지 버전 자산 나열을 참조하세요.

- 자세한 API 내용은 명령 참조GetPackageVersionAsset의 섹션을 참조하세요. AWS CLI

### `get-package-version-readme`

다음 코드 예시에서는 `get-package-version-readme`을 사용하는 방법을 보여 줍니다.

AWS CLI

패키지 버전의 Readme 파일을 가져오려면

다음 `get-package-version-readme` 예제에서는 test-package라는 npm 패키지의 버전 4.0.0에 대한 readme 파일을 검색합니다.

```
aws codeartifact get-package-version-readme \
    --domain test-domain \
    --repo test-repo \
    --format npm \
    --package test-package \
    --package-version 4.0.0
```

출력:

```
{
    "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="
}
```

자세한 내용은 AWS CodeArtifact 사용 설명서의 패키지 버전 읽기 파일 보기를 참조하세요.

- 자세한 API 내용은 명령 참조GetPackageVersionReadme의 섹션을 참조하세요. AWS CLI

### get-repository-endpoint

다음 코드 예시에서는 get-repository-endpoint을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리의 URL 엔드포인트를 가져오려면

다음 get-repository-endpoint 예제에서는 test-repo 리포지토리의 npm 엔드포인트를 반환합니다.

```
aws codeartifact get-repository-endpoint \
    --domain test-domain \
    --repository test-repo \
    --format npm
```

출력:

```
{
    "repositoryEndpoint": "https://test-domain-111122223333.d.codeartifact.us-
west-2.amazonaws.com/npm/test-repo/"
}
```

자세한 내용은 AWS CodeArtifact 사용 설명서의 리포지토리에 연결을 참조하세요.

- 자세한 API 내용은 명령 참조GetRepositoryEndpoint의 섹션을 참조하세요. AWS CLI

### get-repository-permissions-policy

다음 코드 예시에서는 get-repository-permissions-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리에 대한 권한 정책 문서를 가져오려면

다음 get-repository-permissions-policy 예제에서는 test-repo라는 리포지토리에 권한 정책을 연결합니다.

```
aws codeartifact get-repository-permissions-policy \
```

```
    --domain test-domain \
    --repository test-repo
```

출력:

```
{
    "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": "*"
        }
    ]
}
```

자세한 내용은 AWS CodeArtifact 사용 설명서의 정책 읽기를 참조하세요.

- 자세한 API 내용은 명령 참조GetRepositoryPermissionsPolicy의 섹션을 참조하세요. AWS CLI

### `list-domains`

다음 코드 예시에서는 `list-domains`을 사용하는 방법을 보여 줍니다.

AWS CLI

도메인을 나열하려면

다음 `list-domains` 예제에서는 호출을 수행하는 AWS 계정이 소유한 모든 도메인의 요약을 반환합니다.

```
aws codeartifact list-domains
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS CodeArtifact 사용 설명서의 에서 도메인 작업을 CodeArtifact 참조하세요.

- 자세한 API 내용은 명령 참조ListDomains의 섹션을 참조하세요. AWS CLI

### `list-package-version-assets`

다음 코드 예시에서는 `list-package-version-assets`을 사용하는 방법을 보여 줍니다.

AWS CLI

패키지 버전의 자산을 보려면

다음 `list-package-version-assets` 예제에서는 test-package라는 npm 패키지의 버전 4.0.0에 대한 자산을 검색합니다.

```
aws codeartifact list-package-version-assets \
    --domain test-domain \
    --repo test-repo \
    --format npm \
```

```
    --package test-package \
    --package-version 4.0.0
```

출력:

```
{
    "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
            }
        }
    ]
}
```

자세한 내용은 AWS CodeArtifact 사용 설명서의 패키지 버전 자산 나열을 참조하세요.

- 자세한 API 내용은 명령 참조ListPackageVersionAssets의 섹션을 참조하세요. AWS CLI

### `list-package-version-dependencies`

다음 코드 예시에서는 `list-package-version-dependencies`을 사용하는 방법을 보여 줍니다.

AWS CLI

패키지 버전의 종속성을 보려면

다음 `list-package-version-dependencies` 예제에서는 test-package라는 npm 패키지의 버전 4.0.0에 대한 종속성을 검색합니다.

```
aws codeartifact list-package-version-dependencies \
    --domain test-domain \
```

```
    --repo test-repo \
    --format npm \
    --package test-package \
    --package-version 4.0.0
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS CodeArtifact 사용 설명서의 패키지 버전 세부 정보 및 종속 항목 보기 및 업데이트를 참조하세요.

- 자세한 API 내용은 명령 참조ListPackageVersionDependencies의 섹션을 참조하세요. AWS CLI

### `list-package-versions`

다음 코드 예시에서는 `list-package-versions`을 사용하는 방법을 보여 줍니다.

AWS CLI

패키지의 패키지 버전을 나열하려면

다음 `list-package-versions` 예제에서는 라는 패키지의 패키지 버전 목록을 반환합니다`kind-of`.

```
aws codeartifact list-package-versions \
    --package kind-of \
    --domain test-domain \
    --repository test-repo \
    --format npm
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS CodeArtifact 사용 설명서의 패키지 버전 나열을 참조하세요.

- 자세한 API 내용은 명령 참조ListPackageVersions의 섹션을 참조하세요. AWS CLI

### `list-packages`

다음 코드 예시에서는 `list-packages`을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리의 패키지를 나열하려면

다음 `list-packages` 예제에서는 라는 도메인에 이름이 지정된 리포지토리`test-repo`의 패키지를 나열합니다`test-domain`.

```
aws codeartifact list-packages \
    --domain test-domain \
    --repository test-repo
```

출력:

```
{
    "packages": [
        {
            "format": "npm",
            "package": "lodash"
        }
        {
            "format": "python",
            "package": "test-package"
        }
    ]
}
```

자세한 내용은 AWS CodeArtifact 사용 설명서의 패키지 이름 나열을 참조하세요.

- 자세한 API 내용은 명령 참조ListPackages의 섹션을 참조하세요. AWS CLI

### `list-repositories-in-domain`

다음 코드 예시에서는 `list-repositories-in-domain`을 사용하는 방법을 보여 줍니다.

AWS CLI

도메인의 리포지토리를 나열하려면

다음 `list-repositories-in-domain` 예제에서는 테스트 도메인 도메인의 모든 리포지토리에 대한 요약을 반환합니다.

```
aws codeartifact list-repositories-in-domain \
    --domain test-domain
```

출력:

```
{
    "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."
        }
    ]
}
```

자세한 내용은 AWS CodeArtifact 사용 설명서의 리포지토리 나열을 참조하세요.

- 자세한 API 내용은 명령 참조ListRepositoriesInDomain의 섹션을 참조하세요. AWS CLI

### `list-repositories`

다음 코드 예시에서는 `list-repositories`을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리를 나열하려면

다음 `list-repositories` 예제에서는 호출을 수행하는 AWS 계정이 소유한 도메인의 모든 리포지토리에 대한 요약을 반환합니다.

```
aws codeartifact list-repositories
```

출력:

```
{
    "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."
        }
    ]
}
```

자세한 내용은 AWS CodeArtifact 사용 설명서의 리포지토리 나열을 참조하세요.

- 자세한 API 내용은 명령 참조ListRepositories의 섹션을 참조하세요. AWS CLI

### `login`

다음 코드 예시에서는 `login`을 사용하는 방법을 보여 줍니다.

AWS CLI

로그인 명령을 사용하여 리포지토리에 대한 인증을 구성하려면

다음 `login` 예제에서는 테스트 도메인이라는 도메인에 test-repo라는 리포지토리를 사용하여 npm 패키지 관리자를 구성합니다.

```
aws codeartifact login \
    --domain test-domain \
    --repository test-repo \
    --tool npm
```

출력:

```
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
```

자세한 내용은 AWS CodeArtifact 사용 설명서의 시작하기 AWS CLI를 참조하세요.

- API 자세한 내용은 AWS CLI 명령 참조의 로그인을 참조하세요.

### `put-domain-permissions-policy`

다음 코드 예시에서는 `put-domain-permissions-policy`을 사용하는 방법을 보여 줍니다.

AWS CLI

도메인에 권한 정책을 연결하려면

다음 `put-domain-permissions-policy` 예제에서는 policy.json 파일에 정의된 권한 정책을 test-domain이라는 도메인에 연결합니다.

```
aws codeartifact put-domain-permissions-policy \
```

```
    --domain test-domain \
    --policy-document file://PATH/TO/policy.json
```

출력:

```
{
    "policy": {
        "resourceArn": "arn:aws:codeartifact:region-id:111122223333:domain/test-
domain",
        "document": "{ ...policy document content...}",
        "revision": "MQlyyTQRASRU3HB58gBtSDHXG7Q3hvxxxxxxx="
    }
}
```

자세한 내용은 AWS CodeArtifact 사용 설명서의 도메인 정책 설정을 참조하세요.

- 자세한 API 내용은 명령 참조PutDomainPermissionsPolicy의 섹션을 참조하세요. AWS CLI

### put-repository-permissions-policy

다음 코드 예시에서는 put-repository-permissions-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리에 권한 정책을 연결하려면

다음 put-repository-permissions-policy 예제에서는 policy.json 파일에 정의된 권한 정책을 test-repo라는 리포지토리에 연결합니다.

```
aws codeartifact put-repository-permissions-policy \
    --domain test-domain \
    --repository test-repo \
    --policy-document file://PATH/TO/policy.json
```

출력:

```
{
    "policy": {
        "resourceArn": "arn:aws:codeartifact:region-id:111122223333:repository/test-
domain/test-repo",
        "document": "{ ...policy document content...}",
```

```
        "revision": "MQlyyTQRASRU3HB58gBtSDHXG7Q3hvxxxxxxx="
    }
}
```

자세한 내용은 AWS CodeArtifact 사용 설명서의 정책 설정을 참조하세요.

- 자세한 API 내용은 명령 참조PutRepositoryPermissionsPolicy의 섹션을 참조하세요. AWS CLI

### update-package-versions-status

다음 코드 예시에서는 update-package-versions-status을 사용하는 방법을 보여 줍니다.

AWS CLI

패키지 버전 상태를 업데이트하려면

다음 update-package-versions-status 예제에서는 테스트 패키지의 버전 4.0.0 상태를 아카이브됨으로 업데이트합니다.

```
aws codeartifact update-package-versions-status \
    --domain test-domain \
    --repo test-repo \
    --format npm \
    --package test-package \
    --versions 4.0.0 \
    --target-status Archived
```

출력:

```
{
    "successfulVersions": {
        "4.0.0": {
            "revision": "Ciqe5/9yicvkJT13b5/LdLpCyE6fqA7poa9qp+FilPs=",
            "status": "Archived"
        }
    },
    "failedVersions": {}
}
```

자세한 내용은 AWS CodeArtifact 사용 설명서의 패키지 버전 상태 업데이트를 참조하세요.

- 자세한 API 내용은 명령 참조UpdatePackageVersionsStatus의 섹션을 참조하세요. AWS CLI

**update-repository**

다음 코드 예시에서는 update-repository을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리를 업데이트하려면

다음 update-repository 예제에서는 test-domain이라는 도메인의 test-repo라는 리포지토리에 대한 설명을 “업데이트된 설명입니다”로 업데이트합니다.

```
aws codeartifact update-repository \
    --domain test-domain \
    --repository test-repo \
    --description "this is an updated description"
```

출력:

```
{
    "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": []
    }
}
```

자세한 내용은 AWS CodeArtifact 사용 설명서의 리포지토리 구성 보기 또는 수정을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateRepository의 섹션을 참조하세요. AWS CLI

## CodeBuild 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 CodeBuild.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### batch-delete-builds

다음 코드 예시에서는 batch-delete-builds을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 빌드를 삭제합니다 AWS CodeBuild.

다음 batch-delete-builds 예제에서는 지정된 를 CodeBuild 사용하여 의 빌드를 삭제합니다 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
```

출력:

```
{
    "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"
    ]
}
```

자세한 내용은 AWS CodeBuild 사용 설명서의 빌드 삭제(AWS CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조BatchDeleteBuilds의 섹션을 참조하세요. AWS CLI

### batch-get-build-batches

다음 코드 예시에서는 batch-get-build-batches을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 빌드의 세부 정보를 봅니다 AWS CodeBuild.

다음 batch-get-build-batches 예제에서는 지정된 를 CodeBuild 사용하여 의 빌드 배치에 대한 정보를 가져옵니다IDs.

```
aws codebuild batch-get-build-batches \
    --ids codebuild-demo-project:e9c4f4df-3f43-41d2-ab3a-60fe2EXAMPLE
```

출력:

```
{
    "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": []
}
```

자세한 내용은 AWS CodeBuild 사용 설명서의 AWS CodeBuild <https://docs.aws.amazon.com/codebuild/latest/userguide/batch-build.html>)__의 배치 빌드를 참조하세요.

- 자세한 API 내용은 명령 참조BatchGetBuildBatches의 섹션을 참조하세요. AWS CLI

### `batch-get-builds`

다음 코드 예시에서는 `batch-get-builds`을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 빌드의 세부 정보를 봅니다 AWS CodeBuild.

다음 `batch-get-builds` 예제에서는 지정된 를 CodeBuild 사용한 의 빌드에 대한 정보를 가져옵니다IDs.

```
aws codebuild batch-get-builds --ids codebuild-demo-project:e9c4f4df-3f43-41d2-ab3a-60fe2EXAMPLE codebuild-demo-project:815e755f-bade-4a7e-80f0-efe51EXAMPLE
```

출력:

```
{
    "buildsNotFound": [],
    "builds": [
        {
            "artifacts": {
                "md5sum": "0e95edf915048a0c22efe6d139fff837",
```

```
                "location": "arn:aws:s3:::codepipeline-us-west-2-820783811474/
CodeBuild-Python-Pip/BuildArtif/6DJsqQa",
                "encryptionDisabled": false,
                "sha256sum":
 "cfa0df33a090966a737f64ae4fe498969fdc842a0c9aec540bf93c37ac0d05a2"
            },
            "logs": {
                "cloudWatchLogs": {
                    "status": "ENABLED"
                },
                "s3Logs": {
                    "status": "DISABLED"
                },
                "streamName": "46472baf-8f6b-43c2-9255-b3b963af2732",
                "groupName": "/aws/codebuild/codebuild-demo-project",
                "deepLink": "https://console.aws.amazon.com/cloudwatch/
home?region=us-west-2#logEvent:group=/aws/codebuild/codebuild-demo-
project;stream=46472baf-8f6b-43c2-9255-b3b963af2732"
            },
            "timeoutInMinutes": 60,
            "environment": {
                "privilegedMode": false,
                "computeType": "BUILD_GENERAL1_MEDIUM",
                "image": "aws/codebuild/windows-base:1.0",
                "environmentVariables": [],
                "type": "WINDOWS_CONTAINER"
            },
            "projectName": "codebuild-demo-project",
            "buildComplete": true,
            "source": {
                "gitCloneDepth": 1,
                "insecureSsl": false,
                "type": "CODEPIPELINE"
            },
            "buildStatus": "SUCCEEDED",
            "secondaryArtifacts": [],
            "phases": [
                {
                    "durationInSeconds": 0,
                    "startTime": 1548717462.122,
                    "phaseType": "SUBMITTED",
                    "endTime": 1548717462.484,
                    "phaseStatus": "SUCCEEDED"
                },
```

```
            {
                "durationInSeconds": 0,
                "startTime": 1548717462.484,
                "phaseType": "QUEUED",
                "endTime": 1548717462.775,
                "phaseStatus": "SUCCEEDED"
            },
            {
                "durationInSeconds": 34,
                "endTime": 1548717496.909,
                "contexts": [
                    {
                        "statusCode": "",
                        "message": ""
                    }
                ],
                "startTime": 1548717462.775,
                "phaseType": "PROVISIONING",
                "phaseStatus": "SUCCEEDED"
            },
            {
                "durationInSeconds": 15,
                "endTime": 1548717512.555,
                "contexts": [
                    {
                        "statusCode": "",
                        "message": ""
                    }
                ],
                "startTime": 1548717496.909,
                "phaseType": "DOWNLOAD_SOURCE",
                "phaseStatus": "SUCCEEDED"
            },
            {
                "durationInSeconds": 0,
                "endTime": 1548717512.734,
                "contexts": [
                    {
                        "statusCode": "",
                        "message": ""
                    }
                ],
                "startTime": 1548717512.555,
                "phaseType": "INSTALL",
```

```
            "phaseStatus": "SUCCEEDED"
        },
        {
            "durationInSeconds": 0,
            "endTime": 1548717512.924,
            "contexts": [
                {
                    "statusCode": "",
                    "message": ""
                }
            ],
            "startTime": 1548717512.734,
            "phaseType": "PRE_BUILD",
            "phaseStatus": "SUCCEEDED"
        },
        {
            "durationInSeconds": 9,
            "endTime": 1548717522.254,
            "contexts": [
                {
                    "statusCode": "",
                    "message": ""
                }
            ],
            "startTime": 1548717512.924,
            "phaseType": "BUILD",
            "phaseStatus": "SUCCEEDED"
        },
        {
            "durationInSeconds": 3,
            "endTime": 1548717525.498,
            "contexts": [
                {
                    "statusCode": "",
                    "message": ""
                }
            ],
            "startTime": 1548717522.254,
            "phaseType": "POST_BUILD",
            "phaseStatus": "SUCCEEDED"
        },
        {
            "durationInSeconds": 9,
            "endTime": 1548717534.646,
```

```
                    "contexts": [
                        {
                            "statusCode": "",
                            "message": ""
                        }
                    ],
                    "startTime": 1548717525.498,
                    "phaseType": "UPLOAD_ARTIFACTS",
                    "phaseStatus": "SUCCEEDED"
                },
                {
                    "durationInSeconds": 2,
                    "endTime": 1548717536.846,
                    "contexts": [
                        {
                            "statusCode": "",
                            "message": ""
                        }
                    ],
                    "startTime": 1548717534.646,
                    "phaseType": "FINALIZING",
                    "phaseStatus": "SUCCEEDED"
                },
                {
                    "startTime": 1548717536.846,
                    "phaseType": "COMPLETED"
                }
            ],
            "startTime": 1548717462.122,
            "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
            "initiator": "codepipeline/CodeBuild-Pipeline",
            "secondarySources": [],
            "serviceRole": "arn:aws:iam::123456789012:role/service-role/my-
codebuild-service-role",
            "currentPhase": "COMPLETED",
            "id": "codebuild-demo-project:e9c4f4df-3f43-41d2-ab3a-60fe2EXAMPLE",
            "cache": {
                "type": "NO_CACHE"
            },
            "sourceVersion": "arn:aws:s3:::codepipeline-us-west-2-820783811474/
CodeBuild-Python-Pip/SourceArti/1TspnN3.zip",
            "endTime": 1548717536.846,
            "arn": "arn:aws:codebuild:us-west-2:123456789012:build/codebuild-demo-
project:e9c4f4df-3f43-41d2-ab3a-60fe2EXAMPLE",
```

```
            "queuedTimeoutInMinutes": 480,
            "resolvedSourceVersion": "f2194c1757bbdcb0f8f229254a4b3c8b27d43e0b"
        },
        {
            "artifacts": {
                "md5sum": "",
                "overrideArtifactName": false,
                "location": "arn:aws:s3:::my-artifacts/codebuild-demo-project",
                "encryptionDisabled": false,
                "sha256sum": ""
            },
            "logs": {
                "cloudWatchLogs": {
                    "status": "ENABLED"
                },
                "s3Logs": {
                    "status": "DISABLED"
                },
                "streamName": "4dea3ca4-20ec-4898-b22a-a9eb9292775d",
                "groupName": "/aws/codebuild/codebuild-demo-project",
                "deepLink": "https://console.aws.amazon.com/cloudwatch/
home?region=us-west-2#logEvent:group=/aws/codebuild/codebuild-demo-
project;stream=4dea3ca4-20ec-4898-b22a-a9eb9292775d"
            },
            "timeoutInMinutes": 60,
            "environment": {
                "privilegedMode": false,
                "computeType": "BUILD_GENERAL1_MEDIUM",
                "image": "aws/codebuild/windows-base:1.0",
                "environmentVariables": [],
                "type": "WINDOWS_CONTAINER"
            },
            "projectName": "codebuild-demo-project",
            "buildComplete": true,
            "source": {
                "gitCloneDepth": 1,
                "location": "https://github.com/my-repo/codebuild-demo-project.git",
                "insecureSsl": false,
                "reportBuildStatus": false,
                "type": "GITHUB"
            },
            "buildStatus": "SUCCEEDED",
            "secondaryArtifacts": [],
            "phases": [
```

```
                {
                    "durationInSeconds": 0,
                    "startTime": 1548716241.89,
                    "phaseType": "SUBMITTED",
                    "endTime": 1548716242.241,
                    "phaseStatus": "SUCCEEDED"
                },
                {
                    "durationInSeconds": 0,
                    "startTime": 1548716242.241,
                    "phaseType": "QUEUED",
                    "endTime": 1548716242.536,
                    "phaseStatus": "SUCCEEDED"
                },
                {
                    "durationInSeconds": 33,
                    "endTime": 1548716276.171,
                    "contexts": [
                        {
                            "statusCode": "",
                            "message": ""
                        }
                    ],
                    "startTime": 1548716242.536,
                    "phaseType": "PROVISIONING",
                    "phaseStatus": "SUCCEEDED"
                },
                {
                    "durationInSeconds": 15,
                    "endTime": 1548716291.809,
                    "contexts": [
                        {
                            "statusCode": "",
                            "message": ""
                        }
                    ],
                    "startTime": 1548716276.171,
                    "phaseType": "DOWNLOAD_SOURCE",
                    "phaseStatus": "SUCCEEDED"
                },
                {
                    "durationInSeconds": 0,
                    "endTime": 1548716291.993,
                    "contexts": [
```

```
                    {
                        "statusCode": "",
                        "message": ""
                    }
                ],
                "startTime": 1548716291.809,
                "phaseType": "INSTALL",
                "phaseStatus": "SUCCEEDED"
            },
            {
                "durationInSeconds": 0,
                "endTime": 1548716292.191,
                "contexts": [
                    {
                        "statusCode": "",
                        "message": ""
                    }
                ],
                "startTime": 1548716291.993,
                "phaseType": "PRE_BUILD",
                "phaseStatus": "SUCCEEDED"
            },
            {
                "durationInSeconds": 9,
                "endTime": 1548716301.622,
                "contexts": [
                    {
                        "statusCode": "",
                        "message": ""
                    }
                ],
                "startTime": 1548716292.191,
                "phaseType": "BUILD",
                "phaseStatus": "SUCCEEDED"
            },
            {
                "durationInSeconds": 3,
                "endTime": 1548716304.783,
                "contexts": [
                    {
                        "statusCode": "",
                        "message": ""
                    }
                ],
```

```
                    "startTime": 1548716301.622,
                    "phaseType": "POST_BUILD",
                    "phaseStatus": "SUCCEEDED"
                },
                {
                    "durationInSeconds": 8,
                    "endTime": 1548716313.775,
                    "contexts": [
                        {
                            "statusCode": "",
                            "message": ""
                        }
                    ],
                    "startTime": 1548716304.783,
                    "phaseType": "UPLOAD_ARTIFACTS",
                    "phaseStatus": "SUCCEEDED"
                },
                {
                    "durationInSeconds": 2,
                    "endTime": 1548716315.935,
                    "contexts": [
                        {
                            "statusCode": "",
                            "message": ""
                        }
                    ],
                    "startTime": 1548716313.775,
                    "phaseType": "FINALIZING",
                    "phaseStatus": "SUCCEEDED"
                },
                {
                    "startTime": 1548716315.935,
                    "phaseType": "COMPLETED"
                }
            ],
            "startTime": 1548716241.89,
            "secondarySourceVersions": [],
            "initiator": "my-codebuild-project",
            "arn": "arn:aws:codebuild:us-west-2:123456789012:build/codebuild-demo-
project:815e755f-bade-4a7e-80f0-efe51EXAMPLE",
            "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
            "serviceRole": "arn:aws:iam::123456789012:role/service-role/my-
codebuild-service-role",
            "currentPhase": "COMPLETED",
```

```
            "id": "codebuild-demo-project:815e755f-bade-4a7e-80f0-efe51EXAMPLE",
            "cache": {
                "type": "NO_CACHE"
            },
            "endTime": 1548716315.935,
            "secondarySources": [],
            "queuedTimeoutInMinutes": 480,
            "resolvedSourceVersion": "f2194c1757bbdcb0f8f229254a4b3c8b27d43e0b"
        }
    ]
}
```

자세한 내용은 AWS CodeBuild 사용 설명서의 빌드 세부 정보 보기(AWS CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조BatchGetBuilds의 섹션을 참조하세요. AWS CLI

### batch-get-projects

다음 코드 예시에서는 batch-get-projects을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS CodeBuild 빌드 프로젝트 이름 목록을 가져옵니다.

다음 batch-get-projects 예제에서는 이름으로 지정된 CodeBuild 빌드 프로젝트 목록을 가져옵니다.

```
aws codebuild batch-get-projects --names codebuild-demo-project codebuild-demo-project2 my-other-demo-project
```

다음 출력에서 projectsNotFound 배열은 지정되었지만 찾을 수 없는 빌드 프로젝트 이름을 나열합니다. projects 배열에는 정보가 발견된 각 빌드 프로젝트의 세부 정보가 나열됩니다.

```
{
    "projectsNotFound": [],
    "projects": [
        {
            "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
            "name": "codebuild-demo-project2",
            "queuedTimeoutInMinutes": 480,
            "timeoutInMinutes": 60,
            "source": {
```

```
                "buildspec": "version: 0.2\n\n#env:\n  #variables:\n     # key:
 \"value\"\n     # key: \"value\"\n  #parameter-store:\n     # key: \"value\"\n
  # key:\"value\"\n\nphases:\n  #install:\n    #commands:\n      # - command\n
  # - command\n  #pre_build:\n    #commands:\n      # - command\n      # - command
\n  build:\n    commands:\n      # - command\n      # - command\n  #post_build:\n
   #commands:\n      # - command\n      # - command\n#artifacts:\n  #files:\n    #
 - location\n    # - location\n  #name: $(date +%Y-%m-%d)\n  #discard-paths: yes\n
 #base-directory: location\n#cache:\n  #paths:\n    # - paths",
                "type": "NO_SOURCE",
                "insecureSsl": false,
                "gitCloneDepth": 1
            },
            "artifacts": {
                "type": "NO_ARTIFACTS"
            },
            "badge": {
                "badgeEnabled": false
            },
            "lastModified": 1540588091.108,
            "created": 1540588091.108,
            "arn": "arn:aws:codebuild:us-west-2:123456789012:project/test-for-
sample",
            "secondarySources": [],
            "secondaryArtifacts": [],
            "cache": {
                "type": "NO_CACHE"
            },
            "serviceRole": "arn:aws:iam::123456789012:role/service-role/my-test-
role",
            "environment": {
                "image": "aws/codebuild/java:openjdk-8",
                "privilegedMode": true,
                "type": "LINUX_CONTAINER",
                "computeType": "BUILD_GENERAL1_SMALL",
                "environmentVariables": []
            },
            "tags": []
        },
        {
            "encryptionKey": "arn:aws:kms:us-west-2:123456789012:alias/aws/s3",
            "name": "my-other-demo-project",
            "queuedTimeoutInMinutes": 480,
            "timeoutInMinutes": 60,
            "source": {
```

```
                "location": "https://github.com/iversonic/codedeploy-sample.git",
                "reportBuildStatus": false,
                "buildspec": "buildspec.yml",
                "insecureSsl": false,
                "gitCloneDepth": 1,
                "type": "GITHUB",
                "auth": {
                    "type": "OAUTH"
                }
            },
            "artifacts": {
                "type": "NO_ARTIFACTS"
            },
            "badge": {
                "badgeEnabled": false
            },
            "lastModified": 1523401711.73,
            "created": 1523401711.73,
            "arn": "arn:aws:codebuild:us-west-2:123456789012:project/Project2",
            "cache": {
                "type": "NO_CACHE"
            },
            "serviceRole": "arn:aws:iam::123456789012:role/service-role/codebuild-
Project2-service-role",
            "environment": {
                "image": "aws/codebuild/nodejs:4.4.7",
                "privilegedMode": false,
                "type": "LINUX_CONTAINER",
                "computeType": "BUILD_GENERAL1_SMALL",
                "environmentVariables": []
            },
            "tags": []
        }
    ]
}
```

자세한 내용은 AWS CodeBuild 사용 설명서의 빌드 프로젝트 세부 정보 보기(AWS CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조BatchGetProjects의 섹션을 참조하세요. AWS CLI

### batch-get-report-groups

다음 코드 예시에서는 batch-get-report-groups을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 하나 이상의 보고서 그룹에 대한 정보를 가져옵니다 AWS CodeBuild.

다음 batch-get-report-groups 예제에서는 지정된 를 사용하여 보고서 그룹에 대한 정보를 검색합니다ARN.

```
aws codebuild batch-get-report-groups \
    --report-group-arns arn:aws:codebuild:<region-ID>:<user-ID>:report-group/
<report-group-name>
```

출력:

```
{
    "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": []
}
```

자세한 내용은 AWS CodeBuild 사용 설명서의 보고서 그룹 작업을 참조하세요.

- 자세한 API 내용은 명령 참조BatchGetReportGroups의 섹션을 참조하세요. AWS CLI

### batch-get-reports

다음 코드 예시에서는 batch-get-reports을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 하나 이상의 보고서에 대한 정보를 가져옵니다 AWS CodeBuild.

다음 `batch-get-reports` 예제에서는 지정된 를 사용하여 보고서에 대한 정보를 검색합니다 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>
```

출력:

```
{
    "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": []
}
```

자세한 내용은 AWS CodeBuild 사용 설명서의 보고서 작업을 참조하세요.

- 자세한 API 내용은 명령 참조BatchGetReports의 섹션을 참조하세요. AWS CLI

### `create-project`

다음 코드 예시에서는 `create-project`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: AWS CodeBuild 빌드 프로젝트 생성

다음 create-project 예제에서는 S3 버킷의 소스 파일을 사용하여 CodeBuild 빌드 프로젝트를 생성합니다.

```
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"
```

출력:

```
{
    "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
    }
}
```

예제 2: 파라미터에 대한 JSON 입력 파일을 사용하여 AWS CodeBuild 빌드 프로젝트를 생성하려면

다음 `create-project` 예제에서는 JSON 입력 파일에 필요한 모든 파라미터를 전달하여 CodeBuild 빌드 프로젝트를 생성합니다. `--generate-cli-skeleton parameter`만 포함하여 명령을 실행하여 입력 파일 템플릿을 생성합니다.

```
aws codebuild create-project --cli-input-json file://create-project.json
```

입력 JSON 파일에는 다음 콘텐츠가 `create-project.json` 포함됩니다.

```
{
    "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"
```

```
}
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS CodeBuild 사용 설명서의 빌드 프로젝트 생성(AWS CLI)을 참조하세요.

- 자세한 API 내용은 명령 참조CreateProject의 섹션을 참조하세요. AWS CLI

### `create-report-group`

다음 코드 예시에서는 `create-report-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 보고서 그룹을 생성합니다 AWS CodeBuild.

다음 `create-report-group` 예제에서는 새 보고서 그룹을 생성합니다.

```
aws codebuild create-report-group \
    --cli-input-json file://create-report-group-source.json
```

create-report-group-source.json의 내용:

```
{
    "name": "cli-created-report-group",
    "type": "TEST",
    "exportConfig": {
        "exportConfigType": "S3",
        "s3Destination": {
            "bucket": "my-s3-bucket",
            "path": "",
            "packaging": "ZIP",
            "encryptionDisabled": true
        }
    }
}
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 AWS CodeBuild 사용 설명서의 보고서 그룹 작업을 참조하세요.

- 자세한 API 내용은 명령 참조CreateReportGroup의 섹션을 참조하세요. AWS CLI

### `create-webhook`

다음 코드 예시에서는 `create-webhook`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS CodeBuild 프로젝트에 대한 웹후크 필터를 생성하려면

다음 `create-webhook` 예제에서는 두 개의 필터 그룹이 `my-project` 있는 라는 CodeBuild 프로젝트에 대한 웹후크를 생성합니다. 첫 번째 필터 그룹은 정규식 `^refs/heads/master$`와 일치하는 Git 참조 이름과 `^refs/heads/myBranch$`와 일치하는 헤드 참조를 갖는 브랜치에서 생성되거나 업데이트되거나 다시 열린 pull 요청을 지정합니다. 두 번째 필터 그룹은 정규식 과 일치하지 않는 Git 참조 이름이 있는 브랜치에 대한 푸시 요청을 지정합니다`^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}]]"
```

출력:

```
{
    "webhook": {
        "payloadUrl": "https://codebuild.us-west-2.amazonaws.com/webhooks?
t=eyJlbmNyeXB0ZWREYXRhIjoiVVl5MGtoeGRwSzZFRXl2Wnh4bld1Z0tKZ291TVpQNEtFamQ3RDlDYWpRaGIreVFrdm
        "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
                }
            ]
        ]
    }
}
```

자세한 내용은 AWS CodeBuild 사용 설명서의 GitHub Webhook 이벤트 필터링(SDK)을 참조하세요.

- 자세한 API 내용은 명령 참조CreateWebhook의 섹션을 참조하세요. AWS CLI

### `delete-build-batch`

다음 코드 예시에서는 `delete-build-batch`을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 배치 빌드를 삭제합니다 AWS CodeBuild.

다음 delete-build-batch 예제에서는 지정된 배치 빌드를 삭제합니다.

```
aws codebuild delete-build-batch \
    --id <project-name>:<batch-ID>
```

출력:

```
{
    "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": []
}
```

자세한 내용은 AWS CodeBuild 사용 설명서의 의 배치 빌드 AWS CodeBuild를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteBuildBatch의 섹션을 참조하세요. AWS CLI

## delete-project

다음 코드 예시에서는 delete-project을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS CodeBuild 빌드 프로젝트를 삭제하려면

다음 delete-project 예제에서는 지정된 CodeBuild 빌드 프로젝트를 삭제합니다.

```
aws codebuild delete-project --name my-project
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS CodeBuild 사용 설명서의 빌드 프로젝트 삭제(AWS CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteProject의 섹션을 참조하세요. AWS CLI

### delete-report-group

다음 코드 예시에서는 delete-report-group을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 보고서 그룹을 삭제합니다 AWS CodeBuild.

다음 delete-report-group 예제에서는 지정된 가 있는 보고서 그룹을 삭제합니다ARN.

```
aws codebuild delete-report-group \
    --arn arn:aws:codebuild:<region-ID>:<user-ID>:report-group/<report-group-name>
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS CodeBuild 사용 설명서의 보고서 그룹 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteReportGroup의 섹션을 참조하세요. AWS CLI

### delete-report

다음 코드 예시에서는 delete-report을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 보고서를 삭제합니다 AWS CodeBuild.

다음 delete-report 예제에서는 지정된 보고서를 삭제합니다.

```
aws codebuild delete-report \
    --arn arn:aws:codebuild:<region-ID>:<account-ID>:report/<report-group-
name>:<report-ID>
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS CodeBuild 사용 설명서의 보고서 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteReport의 섹션을 참조하세요. AWS CLI

### delete-source-credentials

다음 코드 예시에서는 delete-source-credentials을 사용하는 방법을 보여 줍니다.

AWS CLI

소스 공급자와의 연결을 끊고 액세스 토큰을 제거합니다.

다음 delete-source-credentials 예제에서는 소스 공급자의 연결을 해제하고 토큰을 제거합니다. 소스 공급자에 연결하는 데 사용되는 소스 자격 증명ARN의 에 따라 소스 자격 증명이 결정됩니다.

```
aws codebuild delete-source-credentials --arn arn-of-your-credentials
```

출력:

```
{
    "arn": "arn:aws:codebuild:your-region:your-account-id:token/your-server-type"
}
```

자세한 내용은 AWS CodeBuild 사용 설명서의 액세스 토큰(CLI)을 사용하여 소스 공급자 연결을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteSourceCredentials의 섹션을 참조하세요. AWS CLI

### delete-webhook

다음 코드 예시에서는 delete-webhook을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS CodeBuild 프로젝트에서 웹후크 필터를 삭제하려면

다음 delete-webhook 예제에서는 지정된 CodeBuild 프로젝트에서 웹후크를 삭제합니다.

```
aws codebuild delete-webhook --project-name my-project
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS CodeBuild 사용 설명서의 빌드 자동 실행 중지(AWS CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteWebhook의 섹션을 참조하세요. AWS CLI

### describe-code-coverages

다음 코드 예시에서는 describe-code-coverages을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 코드 적용 범위 테스트 결과에 대한 자세한 정보를 가져옵니다 AWS CodeBuild.

다음 describe-code-coverages 예제에서는 지정된 보고서의 코드 적용 범위 테스트 결과에 대한 정보를 가져옵니다.

```
aws codebuild describe-code-coverages \
    --report-arn arn:aws:codebuild:<region-ID>:<account-ID>:report/<report-group-name>:<report-ID>
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS CodeBuild 사용 설명서의 코드 적용 범위 보고서를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeCodeCoverages의 섹션을 참조하세요. AWS CLI

### describe-test-cases

다음 코드 예시에서는 describe-test-cases을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 테스트 사례에 대한 자세한 정보를 가져옵니다 AWS CodeBuild.

다음 describe-test-cases 예제에서는 지정된 보고서의 테스트 사례에 대한 정보를 가져옵니다.

```
aws codebuild describe-test-cases \
    --report-arn arn:aws:codebuild:<region-ID>:<account-ID>:report/<report-group-name>:<report-ID>
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS CodeBuild 사용 설명서의 에서 테스트 보고 작업을 AWS CodeBuild 참조하세요.

- 자세한 API 내용은 명령 참조DescribeTestCases의 섹션을 참조하세요. AWS CLI

### `import-source-credentials`

다음 코드 예시에서는 `import-source-credentials`을 사용하는 방법을 보여 줍니다.

AWS CLI

소스 공급자의 자격 증명을 가져와 AWS CodeBuild 사용자를 소스 공급자에 연결합니다.

다음 `import-source-credentials` 예제에서는 인증 유형에 BASIC_AUTH를 사용하는 Bitbucket 리포지토리의 토큰을 가져옵니다.

```
aws codebuild import-source-credentials --server-type BITBUCKET --auth-
type BASIC_AUTH --token my-Bitbucket-password --username my-Bitbucket-username
```

출력:

```
{
    "arn": "arn:aws:codebuild:us-west-2:123456789012:token/bitbucket"
}
```

자세한 내용은 AWS CodeBuild 사용 설명서의 액세스 토큰(CLI)을 사용하여 소스 공급자 연결을 참조하세요.

- 자세한 API 내용은 명령 참조ImportSourceCredentials의 섹션을 참조하세요. AWS CLI

### invalidate-project-cache

다음 코드 예시에서는 invalidate-project-cache을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS CodeBuild 빌드 프로젝트의 캐시를 재설정합니다.

다음 invalidate-project-cache 예제에서는 지정된 CodeBuild 프로젝트의 캐시를 재설정합니다.

```
aws codebuild invalidate-project-cache --project-name my-project
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS CodeBuild 사용 설명서의 에서 빌드 캐싱 CodeBuild을 참조하세요.

- 자세한 API 내용은 명령 참조InvalidateProjectCache의 섹션을 참조하세요. AWS CLI

### list-build-batches-for-project

다음 코드 예시에서는 list-build-batches-for-project을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 특정 빌드 프로젝트의 배치 빌드를 나열합니다 AWS CodeBuild.

다음 list-build-batches-for-project 예제에서는 지정된 프로젝트의 CodeBuild 배치 빌드를 나열합니다.

```
aws codebuild list-build-batches-for-project \
    --project-name "<project-name>"
```

출력:

```
{
    "ids": [
        "<project-name>:<batch-ID>",
        "<project-name>:<batch-ID>"
    ]
}
```

자세한 내용은 AWS CodeBuild 사용 설명서의 의 배치 빌드 AWS CodeBuild를 참조하세요.

- 자세한 API 내용은 명령 참조ListBuildBatchesForProject의 섹션을 참조하세요. AWS CLI

### `list-build-batches`

다음 코드 예시에서는 `list-build-batches`을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 배치 빌드를 나열합니다 AWS CodeBuild.

다음 `list-build-batches` 예제에서는 현재 계정의 CodeBuild 배치 빌드를 나열합니다.

```
aws codebuild list-build-batches
```

출력:

```
{
    "ids": [
        "<project-name>:<batch-ID>",
        "<project-name>:<batch-ID>"
    ]
}
```

자세한 내용은 AWS CodeBuild 사용 설명서의 AWS CodeBuild <https://docs.aws.amazon.com/codebuild/latest/userguide/batch-build.html>)__의 배치 빌드를 참조하세요.

- 자세한 API 내용은 명령 참조ListBuildBatches의 섹션을 참조하세요. AWS CLI

### `list-builds-for-project`

다음 코드 예시에서는 `list-builds-for-project`을 사용하는 방법을 보여 줍니다.

AWS CLI

빌드 프로젝트의 AWS CodeBuild 빌드 목록을 봅니다.

다음 `list-builds-for-project` 예제에서는 지정된 빌드 프로젝트의 CodeBuild 빌드를 IDs 내림차순으로 나열합니다.

```
aws codebuild list-builds-for-project --project-name codebuild-demo-project --sort-order DESCENDING
```

출력:

```
{
    "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"
    ]
}
```

자세한 내용은 AWS CodeBuild 사용 설명서의 빌드 프로젝트용 빌드 목록 보기(AWS CLI)를 참조 IDs하세요.

- 자세한 API 내용은 명령 참조ListBuildsForProject의 섹션을 참조하세요. AWS CLI

### `list-builds`

다음 코드 예시에서는 `list-builds`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS CodeBuild 빌드 목록을 가져옵니다IDs.

다음 `list-builds` 예제에서는 오름차순으로 정렬된 목록을 CodeBuild IDs 가져옵니다.

```
aws codebuild list-builds --sort-order ASCENDING
```

출력에는 사용 가능한 출력이 더 많음을 나타내는 `nextToken` 값이 포함됩니다.

```
{
    "nextToken": "4AEA6u7J...The full token has been omitted for
 brevity...MzY2OA==",
    "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"
    ]
}
```

이 명령을 다시 실행하고 이전 응답의 nextToken 값을 파라미터로 제공하여 출력의 다음 부분을 가져옵니다. 응답에서 nextToken 값을 받지 못할 때까지 반복합니다.

```
aws codebuild list-builds --sort-order ASCENDING --next-
token 4AEA6u7J...The full token has been omitted for brevity...MzY2OA==
```

출력의 다음 부분:

```
{
    "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"
    ]
}
```

자세한 내용은 AWS CodeBuild 사용 설명서의 빌드 목록 보기IDs(AWS CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조ListBuilds의 섹션을 참조하세요. AWS CLI

### list-curated-environment-images

다음 코드 예시에서는 list-curated-environment-images을 사용하는 방법을 보여 줍니다.

AWS CLI

빌드에 사용할 수 AWS CodeBuild 있는 에서 관리하는 Docker 이미지 목록을 가져옵니다.

다음 list-curated-environment-images 예제에서는 빌드에 사용할 수 CodeBuild 있는 에서 관리하는 Docker 이미지를 나열합니다.

```
aws codebuild list-curated-environment-images
```

출력:

```
{
    "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 ...
                    ]
                }
            ]
        }
    ]
}
```

자세한 내용은 AWS CodeBuild 사용 설명서의 에서 제공하는 Docker 이미지를 CodeBuild 참조하세요.

- 자세한 API 내용은 명령 참조ListCuratedEnvironmentImages의 섹션을 참조하세요. AWS CLI

### `list-projects`

다음 코드 예시에서는 `list-projects`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS CodeBuild 빌드 프로젝트 이름 목록을 가져옵니다.

다음 `list-projects` 예제에서는 이름별로 정렬된 CodeBuild 빌드 프로젝트 목록을 오름차순으로 가져옵니다.

```
aws codebuild list-projects --sort-by NAME --sort-order ASCENDING
```

출력에는 사용 가능한 출력이 더 많음을 나타내는 nextToken 값이 포함됩니다.

```
{
    "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"
    ]
}
```

이 명령을 다시 실행하고 이전 응답의 nextToken 값을 파라미터로 제공하여 출력의 다음 부분을 가져옵니다. 응답에서 nextToken 값을 받지 못할 때까지 반복합니다.

```
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"

  ]
}
```

자세한 내용은 AWS CodeBuild 사용 설명서의 빌드 프로젝트 이름 목록 보기(AWS CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조ListProjects의 섹션을 참조하세요. AWS CLI

### list-report-groups

다음 코드 예시에서는 list-report-groups을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 보고서 그룹 목록을 가져옵니다ARNs AWS CodeBuild.

다음 `list-report-groups` 예제에서는 리전의 계정에 ARNs 대한 보고서 그룹을 검색합니다.

```
aws codebuild list-report-groups
```

출력:

```
{
    "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"
    ]
}
```

자세한 내용은 AWS CodeBuild 사용 설명서의 보고서 그룹 작업을 참조하세요.

- 자세한 API 내용은 명령 참조ListReportGroups의 섹션을 참조하세요. AWS CLI

### `list-reports-for-report-group`

다음 코드 예시에서는 `list-reports-for-report-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 보고서 그룹의 보고서 목록을 가져옵니다 AWS CodeBuild.

다음 `list-report-for-report-groups` 예제에서는 리전의 계정에 대해 지정된 보고서 그룹의 보고서를 검색합니다.

```
aws codebuild list-reports-for-report-group \
    --report-group-arn arn:aws:codebuild:<region-ID>:<user-ID>:report-group/<report-group-name>
```

출력:

```
{
    "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"
    ]
}
```

자세한 내용은 AWS CodeBuild 사용 설명서의 보고서 그룹 작업을 참조하세요.

- 자세한 API 내용은 명령 참조ListReportsForReportGroup의 섹션을 참조하세요. AWS CLI

### `list-reports`

다음 코드 예시에서는 `list-reports`을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 현재 계정에 대한 보고서 목록을 가져옵니다 AWS CodeBuild.

다음 `list-reports` 예제에서는 현재 계정에 대한 보고서의 ARNs 를 검색합니다.

```
aws codebuild list-reports
```

출력:

```
{
    "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>"
    ]
}
```

자세한 내용은 AWS CodeBuild 사용 설명서의 보고서 작업을 참조하세요.

- 자세한 API 내용은 명령 참조ListReports의 섹션을 참조하세요. AWS CLI

### `list-shared-projects`

다음 코드 예시에서는 `list-shared-projects`을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 공유 프로젝트를 나열합니다 AWS CodeBuild.

다음 `list-shared-projects` 예제에서는 현재 계정에서 사용할 수 있는 CodeBuild 공유 프로젝트를 나열합니다.

```
aws codebuild list-shared-projects
```

출력:

```
{
    "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>"
    ]
}
```

자세한 내용은 AWS CodeBuild 사용 설명서의 공유 프로젝트 작업을 참조하세요.

- 자세한 API 내용은 명령 참조ListSharedProjects의 섹션을 참조하세요. AWS CLI

### `list-shared-report-groups`

다음 코드 예시에서는 `list-shared-report-groups`을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 공유 보고서 그룹 목록을 가져옵니다ARNs AWS CodeBuild.

다음 `list-shared-report-groups` 예제에서는 리전의 계정에 ARNs 대한 보고서 그룹을 검색합니다.

```
aws codebuild list-shared-report-groups
```

출력:

```
{
    "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"
    ]
}
```

자세한 내용은 AWS CodeBuild 사용 설명서의 보고서 그룹 작업을 참조하세요.

- 자세한 API 내용은 명령 참조ListSharedReportGroups의 섹션을 참조하세요. AWS CLI

### `list-source-credentials`

다음 코드 예시에서는 `list-source-credentials`을 사용하는 방법을 보여 줍니다.

AWS CLI

의 목록을 보려면 sourceCredentialsObjects

다음 `list-source-credentials` 예제에서는 하나의 Bitbucket AWS 계정과 하나의 GitHub 계정에 연결된 계정의 토큰을 나열합니다. 응답의 각 `sourceCredentialsInfos` 객체에는 연결된 소스 보안 인증 정보가 포함됩니다.

```
aws codebuild list-source-credentials
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS CodeBuild 사용 설명서의 액세스 토큰(CLI)으로 소스 공급자 연결을 참조하세요.

- 자세한 API 내용은 명령 참조ListSourceCredentials의 섹션을 참조하세요. AWS CLI

### retry-build-batch

다음 코드 예시에서는 retry-build-batch을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 실패한 배치 빌드를 재시도합니다 AWS CodeBuild.

다음 retry-build-batch 예제에서는 지정된 배치 빌드를 다시 시작합니다.

```
aws codebuild retry-build-batch \
    --id <project-name>:<batch-ID>
```

출력:

```
{
    "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"
                }
            }
        ]
    }
}
```

자세한 내용은 AWS CodeBuild 사용 설명서의 의 배치 빌드 AWS CodeBuild를 참조하세요.

- 자세한 API 내용은 명령 참조RetryBuildBatch의 섹션을 참조하세요. AWS CLI

### retry-build

다음 코드 예시에서는 retry-build을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 실패한 빌드를 다시 시도하려면 AWS CodeBuild

다음 retry-build 예제에서는 지정된 빌드를 다시 시작합니다.

```
aws codebuild retry-build \
    --id <project-name>:<build-ID>
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS CodeBuild 사용 설명서의 의 배치 빌드 AWS CodeBuild를 참조하세요.

- 자세한 API 내용은 명령 참조RetryBuild의 섹션을 참조하세요. AWS CLI

### start-build-batch

다음 코드 예시에서는 start-build-batch을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 배치 빌드를 시작합니다 AWS CodeBuild.

다음 start-build-batch 예제에서는 지정된 프로젝트의 배치 빌드를 시작합니다.

```
aws codebuild start-build-batch \
    --project-name <project-name>
```

출력:

```
{
    "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
        }
    }
}
```

자세한 내용은 AWS CodeBuild 사용 설명서의 의 배치 빌드 AWS CodeBuild를 참조하세요.

- 자세한 API 내용은 명령 참조StartBuildBatch의 섹션을 참조하세요. AWS CLI

### start-build

다음 코드 예시에서는 start-build을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS CodeBuild 빌드 프로젝트 빌드 실행을 시작합니다.

다음 start-build 예제에서는 지정된 CodeBuild 프로젝트에 대한 빌드를 시작합니다. 빌드는 시간이 초과되기 전에 빌드가 대기열에 대기할 수 있는 분 수와 프로젝트의 아티팩트 설정 모두에 대해 프로젝트의 설정을 재정의합니다.

```
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"}
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS CodeBuild 사용 설명서의 빌드 실행(AWS CLI)을 참조하세요.

- 자세한 API 내용은 명령 참조StartBuild의 섹션을 참조하세요. AWS CLI

### stop-build-batch

다음 코드 예시에서는 stop-build-batch을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 진행 중인 배치 빌드를 중지합니다 AWS CodeBuild.

다음 stop-build-batch 예제에서는 지정된 배치 빌드를 중지합니다.

```
aws codebuild stop-build-batch \
    --id <project-name>:<batch-ID>
```

출력:

```
{
    "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"
                }
            }
        ]
    }
}
```

자세한 내용은 AWS CodeBuild 사용 설명서의 의 배치 빌드 AWS CodeBuild를 참조하세요.

- 자세한 API 내용은 명령 참조StopBuildBatch의 섹션을 참조하세요. AWS CLI

### stop-build

다음 코드 예시에서는 stop-build을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS CodeBuild 빌드 프로젝트의 빌드를 중지합니다.

다음 stop-build 예제에서는 지정된 CodeBuild 빌드를 중지합니다.

```
aws codebuild stop-build --id my-demo-project:12345678-a1b2-c3d4-e5f6-11111EXAMPLE
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS CodeBuild 사용 설명서의 빌드 중지(AWS CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조StopBuild의 섹션을 참조하세요. AWS CLI

### update-project

다음 코드 예시에서는 update-project을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS CodeBuild 빌드 프로젝트의 설정을 변경합니다.

다음 update-project 예제에서는 라는 지정된 CodeBuild 빌드 프로젝트의 설정을 변경합니다 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"
```

출력에 업데이트된 설정이 표시됩니다.

```
{
    "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"
        }
    }
}
```

자세한 내용은 AWS CodeBuild 사용 설명서의 빌드 프로젝트 설정 변경(AWS CLI)을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateProject의 섹션을 참조하세요. AWS CLI

### update-report-group

다음 코드 예시에서는 update-report-group을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 보고서 그룹을 업데이트합니다 AWS CodeBuild.

다음 update-report-group 예제에서는 보고서 그룹의 내보내기 유형을 “NO_EXPORT”로 변경합니다.

```
aws codebuild update-report-group \
    --arn arn:aws:codebuild:<region-ID>:<user-ID>:report-group/cli-created-report-
group \
```

```
    --export-config="exportConfigType=NO_EXPORT"
```

출력:

```
{
    "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": []
    }
}
```

자세한 내용은 AWS CodeBuild 사용 설명서의 보고서 그룹 작업을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateReportGroup의 섹션을 참조하세요. AWS CLI

### update-webhook

다음 코드 예시에서는 update-webhook을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS CodeBuild 프로젝트의 웹후크를 업데이트하려면

다음 update-webhook 예제에서는 지정된 CodeBuild 프로젝트의 웹후크를 두 개의 필터 그룹으로 업데이트합니다. --rotate-secret 파라미터는 코드 변경이 빌드를 트리거할 때마다 프로젝트의 보안 키를 GitHub 교체하도록 지정합니다. 첫 번째 필터 그룹은 정규식 ^refs/heads/master$와 일치하는 Git 참조 이름과 ^refs/heads/myBranch$와 일치하는 헤드 참조를 갖는 브랜치에서 생성되거나 업데이트되거나 다시 열린 pull 요청을 지정합니다. 두 번째 필터 그룹은 정규식 과 일치하지 않는 Git 참조 이름이 있는 브랜치에 대한 푸시 요청을 지정합니다^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}]]"
```

출력:

```
{
    "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
```

```
    }
}
```

자세한 내용은 AWS CodeBuild 사용 설명서의 빌드 프로젝트 설정 변경(AWS CLI)을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateWebhook의 섹션을 참조하세요. AWS CLI

## CodeCommit 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 CodeCommit.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `associate-approval-rule-template-with-repository`

다음 코드 예시에서는 `associate-approval-rule-template-with-repository`을 사용하는 방법을 보여 줍니다.

AWS CLI

승인 규칙 템플릿을 리포지토리에 연결하려면

다음 `associate-approval-rule-template-with-repository` 예제에서는 지정된 승인 규칙 템플릿을 라는 리포지토리와 연결합니다MyDemoRepo.

```
aws codecommit associate-approval-rule-template-with-repository \
    --repository-name MyDemoRepo  \
    --approval-rule-template-name 2-approver-rule-for-main
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS CodeCommit 사용 설명서의 승인 규칙 템플릿을 리포지토리와 연결을 참조하세요.

- 자세한 API 내용은 명령 참조AssociateApprovalRuleTemplateWithRepository의 섹션을 참조하세요. AWS CLI

### batch-associate-approval-rule-template-with-repositories

다음 코드 예시에서는 batch-associate-approval-rule-template-with-repositories을 사용하는 방법을 보여 줍니다.

AWS CLI

단일 작업에서 승인 규칙 템플릿을 여러 리포지토리와 연결하려면

다음 batch-associate-approval-rule-template-with-repositories 예제에서는 지정된 승인 규칙 템플릿을 MyDemoRepo 및 라는 리포지토리와 연결합니다MyOtherDemoRepo.

참고: 승인 규칙 템플릿은 템플릿이 생성된 AWS 리전에 따라 다릅니다. 해당 AWS 리전의 리포지토리에만 연결할 수 있습니다.

```
aws codecommit batch-associate-approval-rule-template-with-repositories \
    --repository-names MyDemoRepo, MyOtherDemoRepo  \
    --approval-rule-template-name 2-approver-rule-for-main
```

출력:

```
{
    "associatedRepositoryNames": [
        "MyDemoRepo",
        "MyOtherDemoRepo"
    ],
    "errors": []
}
```

자세한 내용은 AWS CodeCommit 사용 설명서의 승인 규칙 템플릿을 리포지토리와 연결을 참조하세요.

- 자세한 API 내용은 명령 참조BatchAssociateApprovalRuleTemplateWithRepositories의 섹션을 참조하세요. AWS CLI

### batch-describe-merge-conflicts

다음 코드 예시에서는 batch-describe-merge-conflicts을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 파일의 병합 충돌 또는 두 커밋 지정자 간의 병합에 있는 파일 하위 집합에 대한 정보를 가져오려면

다음 batch-describe-merge-conflicts 예제에서는 라는 리포지토리의 THREE_WAY_MERGE 전략을 main 사용하여 라는 대상 브랜치feature-randomizationfeature와 라는 소스 브랜치를 병합하기 위한 병합 충돌을 결정합니다MyDemoRepo.

```
aws codecommit batch-describe-merge-conflicts \
    --source-commit-specifier feature-randomizationfeature \
    --destination-commit-specifier main \
    --merge-option THREE_WAY_MERGE \
    --repository-name MyDemoRepo
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS CodeCommit 사용 설명서의 풀 요청에서 충돌 해결을 참조하세요.

- 자세한 API 내용은 명령 참조BatchDescribeMergeConflicts의 섹션을 참조하세요. AWS CLI

### batch-disassociate-approval-rule-template-from-repositories

다음 코드 예시에서는 batch-disassociate-approval-rule-template-from-repositories을 사용하는 방법을 보여 줍니다.

AWS CLI

단일 작업에서 여러 리포지토리에서 승인 규칙 템플릿을 연결 해제하려면

다음 batch-disassociate-approval-rule-template-from-repositories 예제에서는 지정된 승인 규칙 템플릿을 MyDemoRepo 및 라는 리포지토리와 연결 해제합니다MyOtherDemoRepo.

```
aws codecommit batch-disassociate-approval-rule-template-from-repositories \
    --repository-names MyDemoRepo, MyOtherDemoRepo  \
    --approval-rule-template-name 1-approval-rule-for-all pull requests
```

출력:

```
{
    "disassociatedRepositoryNames": [
        "MyDemoRepo",
        "MyOtherDemoRepo"
    ],
    "errors": []
}
```

자세한 내용은 AWS CodeCommit 사용 설명서의 승인 규칙 템플릿 연결 해제를 참조하세요.

- 자세한 API 내용은 명령 참조BatchDisassociateApprovalRuleTemplateFromRepositories의 섹션을 참조하세요. AWS CLI

### batch-get-commits

다음 코드 예시에서는 batch-get-commits을 사용하는 방법을 보여 줍니다.

AWS CLI

여러 커밋에 대한 정보를 보려면

다음 batch-get-commits 예제에서는 지정된 커밋에 대한 세부 정보를 표시합니다.

```
aws codecommit batch-get-commits  \
    --repository-name MyDemoRepo  \
    --commit-ids 317f8570EXAMPLE 4c925148EXAMPLE
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS CodeCommit 사용 설명서의 커밋 세부 정보 보기를 참조하세요.

- 자세한 API 내용은 명령 참조BatchGetCommits의 섹션을 참조하세요. AWS CLI

### batch-get-repositories

다음 코드 예시에서는 batch-get-repositories을 사용하는 방법을 보여 줍니다.

AWS CLI

여러 리포지토리에 대한 세부 정보를 보려면

이 예제에서는 여러 AWS CodeCommit 리포지토리에 대한 세부 정보를 보여줍니다.

```
aws codecommit batch-get-repositories \
    --repository-names MyDemoRepo MyOtherDemoRepo
```

출력:

```
{
    "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": []
}
```

- 자세한 API 내용은 명령 참조BatchGetRepositories의 섹션을 참조하세요. AWS CLI

### create-approval-rule-template

다음 코드 예시에서는 create-approval-rule-template을 사용하는 방법을 보여 줍니다.

AWS CLI

승인 규칙 템플릿을 생성하려면

다음 create-approval-rule-template 예제에서는 풀 요청을 main브랜치에 병합2-approver-rule-for-main ``. The template requires two users who assume the role of ``CodeCommitReview하기 전에 승인하도록 라는 승인 규칙 템플릿을 생성합니다.

```
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/*\"]}]}"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS CodeCommit 사용 설명서의 승인 규칙 템플릿 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateApprovalRuleTemplate의 섹션을 참조하세요. AWS CLI

### `create-branch`

다음 코드 예시에서는 `create-branch`을 사용하는 방법을 보여 줍니다.

AWS CLI

브랜치를 생성하려면

이 예제에서는 AWS CodeCommit 리포지토리에 브랜치를 생성합니다. 이 명령은 오류가 있는 경우에만 출력을 생성합니다.

명령:

```
aws codecommit create-branch --repository-name MyDemoRepo --branch-name MyNewBranch
 --commit-id 317f8570EXAMPLE
```

출력:

```
None.
```

- 자세한 API 내용은 명령 참조CreateBranch의 섹션을 참조하세요. AWS CLI

### `create-commit`

다음 코드 예시에서는 `create-commit`을 사용하는 방법을 보여 줍니다.

AWS CLI

커밋을 생성하려면

다음 `create-commit` 예제에서는 브main랜치MyDemoRepo에 이름이 지정된 리포지토리에 readme.md 파일을 추가하는 리포지토리에 대한 초기 커밋을 생성하는 방법을 보여줍니다.

```
aws codecommit create-commit \
    --repository-name MyDemoRepo \
    --branch-name main \
    --put-files "filePath=readme.md,fileContent='Welcome to our team repository.'"
```

출력:

```
{
    "filesAdded": [
        {
            "blobId": "5e1c309d-EXAMPLE",
            "absolutePath": "readme.md",
            "fileMode": "NORMAL"
        }
    ],
    "commitId": "4df8b524-EXAMPLE",
    "treeId": "55b57003-EXAMPLE",
    "filesDeleted": [],
    "filesUpdated": []
}
```

자세한 내용은 AWS CodeCommit 사용 설명서의 에서 커밋 생성을 AWS CodeCommit 참조하세요.

- 자세한 API 내용은 명령 참조CreateCommit의 섹션을 참조하세요. AWS CLI

### `create-pull-request-approval-rule`

다음 코드 예시에서는 `create-pull-request-approval-rule`을 사용하는 방법을 보여 줍니다.

AWS CLI

풀 요청에 대한 승인 규칙을 생성하려면

다음 `create-pull-request-approval-rule` 예제에서는 지정된 풀 요청에 `Require two approved approvers` 대해 라는 승인 규칙을 생성합니다. 규칙은 승인 풀에 두 개의 승인이

요구되도록 지정합니다. 풀에는 123456789012 AWS 계정CodeCommitReview에서 의 역할을 수임 CodeCommit 하여 에 액세스하는 모든 사용자가 포함됩니다. 또한 동일한 AWS 계정Nikhil_Jayashankar에서 이름이 지정된 IAM 사용자 또는 페더레이션 사용자도 포함됩니다.

```
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/*\"]}]}"
```

출력:

```
{
    "approvalRule": {
        "approvalRuleName": "Require two approved approvers",
        "lastModifiedDate": 1570752871.932,
        "ruleContentSha256": "7c44e6ebEXAMPLE",
        "creationDate": 1570752871.932,
        "approvalRuleId": "aac33506-EXAMPLE",
        "approvalRuleContent": "{\"Version\": \"2018-11-08\",\"Statements\":
 [{\"Type\": \"Approvers\",\"NumberOfApprovalsNeeded\": 2,\"ApprovalPoolMembers
\": [\"CodeCommitApprovers:123456789012:Nikhil_Jayashankar\",
 \"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
        "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major"
    }
}
```

자세한 내용은 AWS CodeCommit 사용 설명서의 승인 규칙 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreatePullRequestApprovalRule의 섹션을 참조하세요. AWS CLI

### `create-pull-request`

다음 코드 예시에서는 `create-pull-request`을 사용하는 방법을 보여 줍니다.

AWS CLI

풀 요청을 생성하려면

다음 `create-pull-request` 예제에서는 'jane-branch' 소스 브랜치를 대상으로 하고 "이라는 리 AWS CodeCommit 포지토리의 기본 브랜치 'main'에 병합될 '화요일까지 이러한 변경 사항을 검토

하세요MyDemoRepo'라는 설명과 함께 'Pronunciation difficulty Analyzer'라는 풀 요청을 생성합니다.

```
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
```

출력:

```
{
    "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
    }
}
```

- 자세한 API 내용은 명령 참조CreatePullRequest의 섹션을 참조하세요. AWS CLI

### `create-repository`

다음 코드 예시에서는 `create-repository`을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리를 만들려면

이 예제에서는 리포지토리를 생성하고 사용자 AWS 계정과 연결합니다.

명령:

```
aws codecommit create-repository --repository-name MyDemoRepo --repository-
description "My demonstration repository"
```

출력:

```
{
    "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"
    }
}
```

- 자세한 API 내용은 명령 참조CreateRepository의 섹션을 참조하세요. AWS CLI

### `create-unreferenced-merge-commit`

다음 코드 예시에서는 `create-unreferenced-merge-commit`을 사용하는 방법을 보여 줍니다.

AWS CLI

두 커밋 지정자를 병합한 결과를 나타내는 참조되지 않은 커밋을 생성하려면

다음 `create-unreferenced-merge-commit` 예제에서는 라는 리포지토리에서 THREE_WAY_MERGE 전략을 `main` 사용하여 라는 대상 브랜치`bugfix-1234`와 라는 소스 브랜치 간의 병합 결과를 나타내는 커밋을 생성합니다`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."
```

출력:

```
{
    "commitId": "4f178133EXAMPLE",
    "treeId": "389765daEXAMPLE"
}
```

자세한 내용은 AWS CodeCommit 사용 설명서의 풀 요청에서 충돌 해결을 참조하세요.

- 자세한 API 내용은 명령 참조CreateUnreferencedMergeCommit의 섹션을 참조하세요. AWS CLI

### `credential-helper`

다음 코드 예시에서는 `credential-helper`을 사용하는 방법을 보여 줍니다.

AWS CLI

와 함께 AWS CLI에 포함된 자격 증명 도우미를 설정하려면 AWS CodeCommit

`credential-helper` 유틸리티는 에서 직접 호출하도록 설계되지 않았습니다 AWS CLI. 대신 로컬 컴퓨터를 설정하기 위한 `git config` 명령과 함께 파라미터로 사용하기 위한 것입니다. 이를 통해 Git은 CodeCommit 리포지토리HTTPS와 상호 작용하기 위해 Git이 와 인증해야 할 때마다 IAM 사용자 자격 증명 또는 Amazon EC2 인스턴스 역할의 및 암호화 서명 버전을 사용할 AWS 수 있습니다.

```
git config --global credential.helper '!aws codecommit credential-helper $@'
git config --global credential.UseHttpPath true
```

출력:

```
[credential]
    helper = !aws codecommit credential-helper $@
    UseHttpPath = true
```

자세한 내용은 AWS CodeCommit 사용 설명서의 다른 방법 AWS CodeCommit 사용을 위한 설정을 참조하세요. 콘텐츠를 주의 깊게 검토한 다음 AWS CodeCommit 사용 설명서의 Linux, macOS 또는 Unix에서 HTTPS 연결 또는 Windows에서 HTTPS 연결 중 하나의 절차를 따릅니다.

- 자세한 API 내용은 명령 참조CredentialHelper의 섹션을 참조하세요. AWS CLI

### `delete-approval-rule-template`

다음 코드 예시에서는 `delete-approval-rule-template`을 사용하는 방법을 보여 줍니다.

AWS CLI

승인 규칙 템플릿을 삭제하려면

다음 `delete-approval-rule-template` 예제에서는 지정된 승인 규칙 템플릿을 삭제합니다.

```
aws codecommit delete-approval-rule-template  \
    --approval-rule-template-name 1-approver-for-all-pull-requests
```

출력:

```
{
```

```
    "approvalRuleTemplateId": "41de97b7-EXAMPLE"
}
```

자세한 내용은 AWS CodeCommit 사용 설명서의 승인 규칙 템플릿 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteApprovalRuleTemplate의 섹션을 참조하세요. AWS CLI

### `delete-branch`

다음 코드 예시에서는 `delete-branch`을 사용하는 방법을 보여 줍니다.

AWS CLI

브랜치를 삭제하려면

이 예제는 AWS CodeCommit 리포지토리에서 브랜치를 삭제하는 방법을 보여줍니다.

명령:

```
aws codecommit delete-branch --repository-name MyDemoRepo --branch-name MyNewBranch
```

출력:

```
{
  "branch": {
      "commitId": "317f8570EXAMPLE",
      "branchName": "MyNewBranch"
  }
}
```

- 자세한 API 내용은 명령 참조DeleteBranch의 섹션을 참조하세요. AWS CLI

### `delete-comment-content`

다음 코드 예시에서는 `delete-comment-content`을 사용하는 방법을 보여 줍니다.

AWS CLI

주석의 내용을 삭제하려면

주석을 생성한 경우에만 주석의 내용을 삭제할 수 있습니다. 이 예제에서는 시스템 생성 ID가 인 주석의 내용을 삭제하는 방법을 보여줍니다`ff30b348EXAMPLEb9aa670f`.

```
aws codecommit delete-comment-content \
    --comment-id ff30b348EXAMPLEb9aa670f
```

출력:

```
{
    "comment": {
        "creationDate": 1508369768.142,
        "deleted": true,
        "lastModifiedDate": 1508369842.278,
        "clientRequestToken": "123Example",
        "commentId": "ff30b348EXAMPLEb9aa670f",
        "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",
        "callerReactions": [],
        "reactionCounts":
        {
            "CLAP" : 1
        }
    }
}
```

- 자세한 API 내용은 명령 참조DeleteCommentContent의 섹션을 참조하세요. AWS CLI

### delete-file

다음 코드 예시에서는 delete-file을 사용하는 방법을 보여 줍니다.

AWS CLI

파일을 삭제하려면

다음 delete-file 예제에서는 라는 리포지토리README.md에서 가장 최근 커밋 IDmain가 인 브랜치c5709475EXAMPLE에서 라는 파일을 삭제하는 방법을 보여줍니다MyDemoRepo.

```
aws codecommit delete-file \
    --repository-name MyDemoRepo \
    --branch-name main \
    --file-path README.md \
    --parent-commit-id c5709475EXAMPLE
```

출력:

```
{
    "blobId":"559b44fEXAMPLE",
    "commitId":"353cf655EXAMPLE",
    "filePath":"README.md",
    "treeId":"6bc824cEXAMPLE"
}
```

자세한 내용은 AWS CodeCommit API 참조 가이드의 에서 파일 편집 또는 삭제 AWS CodeCommit를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteFile의 섹션을 참조하세요. AWS CLI

### delete-pull-request-approval-rule

다음 코드 예시에서는 delete-pull-request-approval-rule을 사용하는 방법을 보여 줍니다.

AWS CLI

풀 요청에 대한 승인 규칙을 삭제하려면

다음 delete-pull-request-approval-rule 예제에서는 지정된 풀 요청에 My Approval Rule 대해 라는 승인 규칙을 삭제합니다.

```
aws codecommit delete-pull-request-approval-rule  \
    --approval-rule-name "My Approval Rule"  \
    --pull-request-id 15
```

출력:

```
{
    "approvalRuleId": "077d8e8a8-EXAMPLE"
}
```

자세한 내용은 AWS CodeCommit 사용 설명서의 승인 규칙 편집 또는 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeletePullRequestApprovalRule의 섹션을 참조하세요. AWS CLI

### delete-repository

다음 코드 예시에서는 delete-repository을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리를 삭제하려면

이 예제에서는 AWS CodeCommit 리포지토리를 삭제하는 방법을 보여줍니다.

명령:

```
aws codecommit delete-repository --repository-name MyDemoRepo
```

출력:

```
{
  "repositoryId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE"
}
```

- 자세한 API 내용은 명령 참조DeleteRepository의 섹션을 참조하세요. AWS CLI

### `describe-merge-conflicts`

다음 코드 예시에서는 `describe-merge-conflicts`을 사용하는 방법을 보여 줍니다.

AWS CLI

병합 충돌에 대한 자세한 정보를 얻으려면

다음 `describe-merge-conflicts` 예제에서는 THREE_WAY_MERGE 전략을 사용하여 지정된 소스 브랜치 및 대상 브랜치`readme.md`에 이름이 지정된 파일의 병합 충돌을 결정합니다.

```
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
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS CodeCommit 사용 설명서의 풀 요청에서 충돌 해결을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeMergeConflicts의 섹션을 참조하세요. AWS CLI

### `describe-pull-request-events`

다음 코드 예시에서는 `describe-pull-request-events`을 사용하는 방법을 보여 줍니다.

AWS CLI

풀 요청에서 이벤트를 보려면

다음 `describe-pull-request-events` 예제에서는 ID가 '8'인 풀 요청에 대한 이벤트를 검색합니다.

```
aws codecommit describe-pull-request-events --pull-request-id 8
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribePullRequestEvents의 섹션을 참조하세요. AWS CLI

### disassociate-approval-rule-template-from-repository

다음 코드 예시에서는 disassociate-approval-rule-template-from-repository을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리에서 승인 규칙 템플릿을 연결 해제하려면

다음 disassociate-approval-rule-template-from-repository 예제에서는 지정된 승인 규칙 템플릿을 라는 리포지토리에서 연결 해제합니다MyDemoRepo.

```
aws codecommit disassociate-approval-rule-template-from-repository \
    --repository-name MyDemoRepo  \
    --approval-rule-template-name 1-approver-rule-for-all-pull-requests
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS CodeCommit 사용 설명서의 승인 규칙 템플릿 연결 해제를 참조하세요.

- 자세한 API 내용은 명령 참조DisassociateApprovalRuleTemplateFromRepository의 섹션을 참조하세요. AWS CLI

### evaluate-pull-request-approval-rules

다음 코드 예시에서는 evaluate-pull-request-approval-rules을 사용하는 방법을 보여 줍니다.

AWS CLI

풀 요청에 모든 승인 규칙이 충족되었는지 평가하려면

다음 evaluate-pull-request-approval-rules 예제에서는 지정된 풀 요청에 대한 승인 규칙의 상태를 평가합니다. 이 예제에서는 풀 요청에 대한 승인 규칙이 충족되지 않았으므로 명령의 출력에 approved 값이 표시됩니다false.

```
aws codecommit evaluate-pull-request-approval-rules \
    --pull-request-id 27  \
    --revision-id 9f29d167EXAMPLE
```

출력:

```
{
    "evaluation": {
        "approved": false,
        "approvalRulesNotSatisfied": [
            "Require two approved approvers"
        ],
        "overridden": false,
        "approvalRulesSatisfied": []
    }
}
```

자세한 내용은 AWS CodeCommit 사용 설명서의 풀 요청 병합을 참조하세요.

- 자세한 API 내용은 명령 참조EvaluatePullRequestApprovalRules의 섹션을 참조하세요. AWS CLI

### get-approval-rule-template

다음 코드 예시에서는 get-approval-rule-template을 사용하는 방법을 보여 줍니다.

AWS CLI

승인 규칙 템플릿의 내용을 가져오려면

다음 get-approval-rule-template 예제에서는 라는 승인 규칙 템플릿의 내용을 가져옵니다1-approver-rule-for-all-pull-requests.

```
aws codecommit get-approval-rule-template \
    --approval-rule-template-name 1-approver-rule-for-all-pull-requests
```

출력:

```
{
    "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."
    }
}
```

자세한 내용은 AWS CodeCommit 사용 설명서의 승인 규칙 템플릿 관리를 참조하세요.

- 자세한 API 내용은 명령 참조GetApprovalRuleTemplate의 섹션을 참조하세요. AWS CLI

### get-blob

다음 코드 예시에서는 get-blob을 사용하는 방법을 보여 줍니다.

AWS CLI

Git Blob 객체에 대한 정보를 보려면

다음 get-blob 예제에서는 "이라는 AWS CodeCommit 리포지토리에서 ID가 '2eb4af3bEXAMPLE'인 Git Blob에 대한 정보를 검색합니다MyDemoRepo.

```
aws codecommit get-blob  --repository-name MyDemoRepo  --blob-id 2eb4af3bEXAMPLE
```

출력:

```
{
    "content": "QSBCaW5hcnkgTGFyToEXAMPLE="
}
```

- 자세한 API 내용은 명령 참조GetBlob의 섹션을 참조하세요. AWS CLI

### get-branch

다음 코드 예시에서는 get-branch을 사용하는 방법을 보여 줍니다.

AWS CLI

브랜치에 대한 정보를 가져오려면

이 예제에서는 AWS CodeCommit 리포지토리의 브랜치에 대한 정보를 가져옵니다.

명령:

```
aws codecommit get-branch --repository-name MyDemoRepo --branch-name MyNewBranch
```

출력:

```
{
  "BranchInfo": {
        "commitID": "317f8570EXAMPLE",
                "branchName": "MyNewBranch"
  }
}
```

- 자세한 API 내용은 명령 참조GetBranch의 섹션을 참조하세요. AWS CLI

### get-comment-reactions

다음 코드 예시에서는 get-comment-reactions을 사용하는 방법을 보여 줍니다.

AWS CLI

의견에 대한 이모티콘 반응을 보려면

다음 get-comment-reactions 예제에서는 ID가 인 주석에 대한 모든 이모티콘 반응을 나열합니다abcd1234EXAMPLEb5678efgh. 쉘의 글꼴이 이모티콘 버전 1.0 표시를 지원하는 경우 emoji에 이모티콘의 출력이 표시됩니다.

```
aws codecommit get-comment-reactions \
    --comment-id abcd1234EXAMPLEb5678efgh
```

출력:

```
{
    "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"
                ]
            }
        ]
    }
}
```

자세한 내용은 AWS CodeCommit 사용 설명서의 에서 커밋에 대한 설명을 AWS CodeCommit 참조하세요.

- 자세한 API 내용은 명령 참조GetCommentReactions의 섹션을 참조하세요. AWS CLI

### get-comment

다음 코드 예시에서는 get-comment을 사용하는 방법을 보여 줍니다.

AWS CLI

설명의 세부 정보를 보려면

이 예제에서는 시스템에서 생성한 주석 ID가 인 주석의 세부 정보를 보는 방법을 보여줍니다ff30b348EXAMPLEb9aa670f.

```
aws codecommit get-comment \
    --comment-id ff30b348EXAMPLEb9aa670f
```

출력:

```
{
    "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
        }
    }
}
```

- 자세한 API 내용은 명령 참조GetComment의 섹션을 참조하세요. AWS CLI

### get-comments-for-compared-commit

다음 코드 예시에서는 get-comments-for-compared-commit을 사용하는 방법을 보여 줍니다.

AWS CLI

커밋에 대한 주석을 보려면

이 예제에서는 라는 리포지토리에서 두 커밋 간의 비교에 대한 설명을 보는 방법을 보여줍니다MyDemoRepo.

```
aws codecommit get-comments-for-compared-commit \
```

```
    --repository-name MyDemoRepo \
    --before-commit-ID 6e147360EXAMPLE \
    --after-commit-id 317f8570EXAMPLE
```

출력:

```
{
    "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"
}
```

- 자세한 API 내용은 명령 참조GetCommentsForComparedCommit의 섹션을 참조하세요. AWS CLI

### get-comments-for-pull-request

다음 코드 예시에서는 get-comments-for-pull-request을 사용하는 방법을 보여 줍니다.

AWS CLI

풀 요청에 대한 설명을 보려면

이 예제에서는 라는 리포지토리에서 풀 요청에 대한 설명을 보는 방법을 보여줍니다MyDemoRepo.

```
aws codecommit get-comments-for-pull-request \
    --repository-name MyDemoRepo \
    --before-commit-ID 317f8570EXAMPLE \
    --after-commit-id 5d036259EXAMPLE
```

출력:

```
{
    "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"
}
```

- 자세한 API 내용은 명령 참조GetCommentsForPullRequest의 섹션을 참조하세요. AWS CLI

### `get-commit`

다음 코드 예시에서는 `get-commit`을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리의 커밋에 대한 정보를 보려면

이 예제는 "이라는 AWS CodeCommit 리포지토리에서 시스템 생성 ID가 '7e9fd3091thisisanexamplethisisanexample1'인 커밋에 대한 세부 정보를 보여줍니다 MyDemoRepo.

명령:

```
aws codecommit get-commit --repository-name MyDemoRepo --commit-
id 7e9fd3091thisisanexamplethisisanexample1
```

출력:

```
{
  "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"
  }
}
```

- 자세한 API 내용은 명령 참조GetCommit의 섹션을 참조하세요. AWS CLI

### get-differences

다음 코드 예시에서는 get-differences을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리의 커밋 지정자의 차이점에 대한 정보를 가져오려면

이 예제는 이름이 인 AWS CodeCommit 리포지토리의 이름이 변경된 폴더에 있는 두 커밋 지정자 (지점, 태그HEAD, 또는 커밋과 같은 기타 정규화된 참조IDs) 간의 변경 사항에 대한 메타데이터 정보 보기를 보여줍니다 MyDemoRepo. 이 예제에는 이러한 옵션을 사용하여 결과를 제한하는 방법을 자세히 설명하기 위해 --, before-commit-specifier--before-path 및 --after-path를 포함하여 필요하지 않은 몇 가지 옵션이 포함되어 있습니다. 응답에는 파일 모드 권한이 포함됩니다.

명령:

```
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
```

출력:

```
{
  "differences": [
      {
          "afterBlob": {
              "path": "blob.txt",
              "blobId": "2eb4af3b1thisisanexamplethisisanexample1",
              "mode": "100644"
          },
          "changeType": "M",
          "beforeBlob": {
              "path": "blob.txt",
              "blobId": "bf7fcf281thisisanexamplethisisanexample1",
              "mode": "100644"
          }
      }
  ]
```

```
}
```

- 자세한 API 내용은 명령 참조GetDifferences의 섹션을 참조하세요. AWS CLI

### get-file

다음 코드 예시에서는 get-file을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS CodeCommit 리포지토리에 있는 파일의 base-64 인코딩 콘텐츠를 가져오려면

다음 get-file 예제에서는 라는 main리포지토리에 이름이 지정된 브랜치README.md에서 이름이 지정된 파일의 base-64 인코딩 콘텐츠를 가져오는 방법을 보여줍니다MyDemoRepo.

```
aws codecommit get-file \
    --repository-name MyDemoRepo \
    --commit-specifier main \
    --file-path README.md
```

출력:

```
{
    "blobId":"559b44fEXAMPLE",
    "commitId":"c5709475EXAMPLE",
    "fileContent":"IyBQaHVzEXAMPLE",
    "filePath":"README.md",
    "fileMode":"NORMAL",
    "fileSize":1563
}
```

자세한 내용은 AWS CodeCommit API 참조 가이드GetFile의 섹션을 참조하세요.

- 자세한 API 내용은 명령 참조GetFile의 섹션을 참조하세요. AWS CLI

### get-folder

다음 코드 예시에서는 get-folder을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS CodeCommit 리포지토리의 폴더 콘텐츠를 가져오려면

다음 get-folder 예제에서는 라는 리포지토리에서 최상위 폴더의 콘텐츠를 가져오는 방법을 보여줍니다MyDemoRepo.

```
aws codecommit get-folder --repository-name MyDemoRepo --folder-path ""
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS CodeCommit API 참조 가이드 GetFolder 의 섹션을 참조하세요.

- 자세한 API 내용은 명령 참조GetFolder의 섹션을 참조하세요. AWS CLI

### get-merge-commit

다음 코드 예시에서는 get-merge-commit을 사용하는 방법을 보여 줍니다.

AWS CLI

병합 커밋에 대한 자세한 정보를 가져오려면

다음 get-merge-commit 예제에서는 라는 리포지토리에서 THREE_WAY_MERGE 전략을 main 사용하여 라는 대상 브랜치bugfix-bug1234와 라는 소스 브랜치의 병합 커밋에 대한 세부 정보를 표시합니다MyDemoRepo.

```
aws codecommit get-merge-commit \
    --source-commit-specifier bugfix-bug1234 \
    --destination-commit-specifier main \
    --merge-option THREE_WAY_MERGE \
    --repository-name MyDemoRepo
```

출력:

```
{
    "sourceCommitId": "c5709475EXAMPLE",
```

```
    "destinationCommitId": "317f8570EXAMPLE",
    "baseCommitId": "fb12a539EXAMPLE",
    "mergeCommitId": "ffc4d608eEXAMPLE"
}
```

자세한 내용은 AWS CodeCommit 사용 설명서의 커밋 세부 정보 보기를 참조하세요.

- 자세한 API 내용은 명령 참조GetMergeCommit의 섹션을 참조하세요. AWS CLI

### get-merge-conflicts

다음 코드 예시에서는 get-merge-conflicts을 사용하는 방법을 보여 줍니다.

AWS CLI

풀 요청에 대한 병합 충돌이 있는지 확인하려면

다음 get-merge-conflicts 예제는 라는 소스 브랜치의 팁feature-randomizationfeature과 라는 리포지토리의 'main'이라는 대상 브랜치 간에 병합 충돌이 있는지 여부를 보여줍니다MyDemoRepo.

```
aws codecommit get-merge-conflicts \
    --repository-name MyDemoRepo \
    --source-commit-specifier feature-randomizationfeature \
    --destination-commit-specifier main \
    --merge-option THREE_WAY_MERGE
```

출력:

```
{
    "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"
            }
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetMergeConflicts의 섹션을 참조하세요. AWS CLI

### get-merge-options

다음 코드 예시에서는 get-merge-options을 사용하는 방법을 보여 줍니다.

AWS CLI

두 개의 지정된 브랜치를 병합하는 데 사용할 수 있는 병합 옵션에 대한 정보를 가져오려면

다음 get-merge-options 예제에서는 라는 소스 브랜치를 라는 리포지토리main에 라는 대상 브랜치bugfix-bug1234와 병합하는 데 사용할 수 있는 병합 옵션을 결정합니다MyDemoRepo.

```
aws codecommit get-merge-options \
    --source-commit-specifier bugfix-bug1234 \
    --destination-commit-specifier main \
```

```
    --repository-name MyDemoRepo
```

출력:

```
{
    "mergeOptions": [
        "FAST_FORWARD_MERGE",
        "SQUASH_MERGE",
        "THREE_WAY_MERGE"
    ],
    "sourceCommitId": "18059494EXAMPLE",
    "destinationCommitId": "ffd3311dEXAMPLE",
    "baseCommitId": "ffd3311dEXAMPLE"
}
```

자세한 내용은 AWS CodeCommit 사용 설명서의 풀 요청에서 충돌 해결을 참조하세요.

- 자세한 API 내용은 명령 참조GetMergeOptions의 섹션을 참조하세요. AWS CLI

### get-pull-request-approval-states

다음 코드 예시에서는 get-pull-request-approval-states을 사용하는 방법을 보여 줍니다.

AWS CLI

풀 요청에 대한 승인을 보려면

다음 get-pull-request-approval-states 예제에서는 지정된 풀 요청에 대한 승인을 반환합니다.

```
aws codecommit get-pull-request-approval-states \
    --pull-request-id 8 \
    --revision-id 9f29d167EXAMPLE
```

출력:

```
{
    "approvals": [
        {
            "userArn": "arn:aws:iam::123456789012:user/Mary_Major",
            "approvalState": "APPROVE"
        }
```

```
    ]
}
```

자세한 내용은 AWS CodeCommit 사용 설명서의 풀 요청 보기를 참조하세요.

- 자세한 API 내용은 명령 참조GetPullRequestApprovalStates의 섹션을 참조하세요. AWS CLI

### get-pull-request-override-state

다음 코드 예시에서는 get-pull-request-override-state을 사용하는 방법을 보여 줍니다.

AWS CLI

풀 요청의 재정의 상태에 대한 정보를 가져오려면

다음 get-pull-request-override-state 예제에서는 지정된 풀 요청에 대한 재정의 상태를 반환합니다. 이 예제에서는 Mary Major라는 사용자가 풀 요청에 대한 승인 규칙을 재정의했으므로 출력은 값을 반환합니다true.:

```
aws codecommit get-pull-request-override-state \
    --pull-request-id 34  \
    --revision-id 9f29d167EXAMPLE
```

출력:

```
{
    "overridden": true,
    "overrider": "arn:aws:iam::123456789012:user/Mary_Major"
}
```

자세한 내용은 AWS CodeCommit 사용 설명서의 풀 요청에 대한 승인 규칙 재정의를 참조하세요.

- 자세한 API 내용은 명령 참조GetPullRequestOverrideState의 섹션을 참조하세요. AWS CLI

### get-pull-request

다음 코드 예시에서는 get-pull-request을 사용하는 방법을 보여 줍니다.

AWS CLI

풀 요청의 세부 정보를 보려면

이 예제에서는 ID가 인 풀 요청에 대한 정보를 보는 방법을 보여줍니다27.

```
aws codecommit get-pull-request \
    --pull-request-id 27
```

출력:

```
{
    "pullRequest": {
        "approvalRules": [
            {
                "approvalRuleContent": "{\"Version\": \"2018-11-08\",\"Statements\":
 [{\"Type\": \"Approvers\",\"NumberOfApprovalsNeeded\": 2,\"ApprovalPoolMembers\":
 [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
                "approvalRuleId": "dd8b17fe-EXAMPLE",
                "approvalRuleName": "2-approver-rule-for-main",
                "creationDate": 1571356106.936,
                "lastModifiedDate": 571356106.936,
                "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
                "ruleContentSha256": "4711b576EXAMPLE"
            }
        ],
        "lastActivityDate": 1562619583.565,
        "pullRequestTargets": [
            {
                "sourceCommit": "ca45e279EXAMPLE",
                "sourceReference": "refs/heads/bugfix-1234",
                "mergeBase": "a99f5ddbEXAMPLE",
                "destinationReference": "refs/heads/main",
                "mergeMetadata": {
                    "isMerged": false
                },
                "destinationCommit": "2abfc6beEXAMPLE",
                "repositoryName": "MyDemoRepo"
            }
        ],
        "revisionId": "e47def21EXAMPLE",
        "title": "Quick fix for bug 1234",
        "authorArn": "arn:aws:iam::123456789012:user/Nikhil_Jayashankar",
        "clientRequestToken": "d8d7612e-EXAMPLE",
        "creationDate": 1562619583.565,
        "pullRequestId": "27",
        "pullRequestStatus": "OPEN"
```

```
    }
}
```

- 자세한 API 내용은 명령 참조GetPullRequest의 섹션을 참조하세요. AWS CLI

### get-repository-triggers

다음 코드 예시에서는 get-repository-triggers을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리의 트리거에 대한 정보를 가져오려면

이 예제에서는 라는 AWS CodeCommit 리포지토리에 대해 구성된 트리거에 대한 세부 정보를 보여 줍니다MyDemoRepo.

```
aws codecommit get-repository-triggers \
    --repository-name MyDemoRepo
```

출력:

```
{
    "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"
            ]
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetRepositoryTriggers의 섹션을 참조하세요. AWS CLI

### get-repository

다음 코드 예시에서는 get-repository을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리에 대한 정보를 가져오려면

이 예제에서는 AWS CodeCommit 리포지토리에 대한 세부 정보를 보여줍니다.

```
aws codecommit get-repository \
    --repository-name MyDemoRepo
```

출력:

```
{
    "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"
    }
}
```

- 자세한 API 내용은 명령 참조GetRepository의 섹션을 참조하세요. AWS CLI

### `list-approval-rule-templates`

다음 코드 예시에서는 `list-approval-rule-templates`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 리전의 모든 승인 규칙 템플릿을 나열하려면

다음 `list-approval-rule-templates` 예제에서는 지정된 리전의 모든 승인 규칙 템플릿을 나열합니다. AWS 리전이 파라미터로 지정되지 않은 경우 명령은 명령을 실행하는 데 사용된 프로파일에 지정된 리전에 AWS CLI 대한 승인 규칙 템플릿을 반환합니다.

```
aws codecommit list-approval-rule-templates \
    --region us-east-2
```

출력:

```
{
    "approvalRuleTemplateNames": [
        "2-approver-rule-for-main",
        "1-approver-rule-for-all-pull-requests"
    ]
}
```

자세한 내용은 AWS CodeCommit 사용 설명서의 승인 규칙 템플릿 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ListApprovalRuleTemplates의 섹션을 참조하세요. AWS CLI

### `list-associated-approval-rule-templates-for-repository`

다음 코드 예시에서는 `list-associated-approval-rule-templates-for-repository`을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리와 연결된 모든 템플릿을 나열하려면

다음 `list-associated-approval-rule-templates-for-repository` 예제에서는 라는 리포지토리와 연결된 모든 승인 규칙 템플릿을 나열합니다MyDemoRepo.

```
aws codecommit list-associated-approval-rule-templates-for-repository \
    --repository-name MyDemoRepo
```

출력:

```
{
  "approvalRuleTemplateNames": [
    "2-approver-rule-for-main",
    "1-approver-rule-for-all-pull-requests"
  ]
}
```

자세한 내용은 AWS CodeCommit 사용 설명서의 승인 규칙 템플릿 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ListAssociatedApprovalRuleTemplatesForRepository의 섹션을 참조하세요. AWS CLI

### `list-branches`

다음 코드 예시에서는 `list-branches`을 사용하는 방법을 보여 줍니다.

AWS CLI

브랜치 이름 목록을 보려면

이 예제에서는 AWS CodeCommit 리포지토리의 모든 브랜치 이름을 나열합니다.

```
aws codecommit list-branches \
    --repository-name MyDemoRepo
```

출력:

```
{
    "branches": [
        "MyNewBranch",
        "main"
    ]
}
```

- 자세한 API 내용은 명령 참조ListBranches의 섹션을 참조하세요. AWS CLI

### list-pull-requests

다음 코드 예시에서는 list-pull-requests을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리에서 풀 요청 목록을 보려면

이 예제에서는 ARN 'arn:aws:iam::111111111111:user/Li_Juan'이 있는 IAM 사용자가 생성한 풀 요청을 나열하고 'CLOSED'이라는 AWS CodeCommit 리포지토리에서 'MyDemoRepo' 상태를 나열하는 방법을 보여줍니다.

```
aws codecommit list-pull-requests --author-arn arn:aws:iam::111111111111:user/
Li_Juan --pull-request-status CLOSED --repository-name MyDemoRepo
```

출력:

```
{
 "nextToken": "",
 "pullRequestIds": ["2","12","16","22","23","35","30","39","47"]
}
```

- 자세한 API 내용은 명령 참조ListPullRequests의 섹션을 참조하세요. AWS CLI

### list-repositories-for-approval-rule-template

다음 코드 예시에서는 list-repositories-for-approval-rule-template을 사용하는 방법을 보여 줍니다.

AWS CLI

템플릿과 연결된 모든 리포지토리를 나열하려면

다음 list-repositories-for-approval-rule-template 예제에서는 지정된 승인 규칙 템플릿과 연결된 모든 리포지토리를 나열합니다.

```
aws codecommit list-repositories-for-approval-rule-template \
    --approval-rule-template-name 2-approver-rule-for-main
```

출력:

```
{
    "repositoryNames": [
        "MyDemoRepo",
        "MyClonedRepo"
    ]
}
```

자세한 내용은 AWS CodeCommit 사용 설명서의 승인 규칙 템플릿 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ListRepositoriesForApprovalRuleTemplate의 섹션을 참조하세요. AWS CLI

### `list-repositories`

다음 코드 예시에서는 `list-repositories`을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리 목록을 보려면

이 예제에서는 사용자 AWS 계정과 연결된 모든 AWS CodeCommit 리포지토리를 나열합니다.

명령:

```
aws codecommit list-repositories
```

출력:

```
{
  "repositories": [
      {
         "repositoryName": "MyDemoRepo"
         "repositoryId": "f7579e13-b83e-4027-aaef-650c0EXAMPLE",
      },
      {
         "repositoryName": "MyOtherDemoRepo"
         "repositoryId": "cfc29ac4-b0cb-44dc-9990-f6f51EXAMPLE"
      }
  ]
}
```

- 자세한 API 내용은 명령 참조ListRepositories의 섹션을 참조하세요. AWS CLI

### list-tags-for-resource

다음 코드 예시에서는 list-tags-for-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리의 AWS 태그를 보려면

다음 list-tags-for-resource 예제에서는 지정된 리포지토리의 태그 키와 태그 값을 나열합니다.

```
aws codecommit list-tags-for-resource \
    --resource-arn arn:aws:codecommit:us-west-2:111111111111:MyDemoRepo
```

출력:

```
{
    "tags": {
        "Status": "Secret",
        "Team": "Saanvi"
    }
}
```

자세한 내용은 AWS CodeCommit 사용 설명서의 리포지토리 태그 보기를 참조하세요.

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### merge-branches-by-fast-forward

다음 코드 예시에서는 merge-branches-by-fast-forward을 사용하는 방법을 보여 줍니다.

AWS CLI

빠른 전달 병합 전략을 사용하여 두 브랜치를 병합하려면

다음 merge-branches-by-fast-forward 예제에서는 지정된 소스 브랜치를 라는 리포지토리의 지정된 대상 브랜치와 병합합니다MyDemoRepo.

```
aws codecommit merge-branches-by-fast-forward \
    --source-commit-specifier bugfix-bug1234 \
    --destination-commit-specifier bugfix-bug1233 \
```

```
    --repository-name MyDemoRepo
```

출력:

```
{
    "commitId": "4f178133EXAMPLE",
    "treeId": "389765daEXAMPLE"
}
```

자세한 내용은 AWS CodeCommit 사용 설명서의 브랜치 비교 및 병합을 참조하세요.

- 자세한 API 내용은 명령 참조MergeBranchesByFastForward의 섹션을 참조하세요. AWS CLI

### merge-branches-by-squash

다음 코드 예시에서는 merge-branches-by-squash을 사용하는 방법을 보여 줍니다.

AWS CLI

스쿼시 병합 전략을 사용하여 두 브랜치를 병합하려면

다음 merge-branches-by-squash 예제에서는 지정된 소스 브랜치를 라는 리포지토리의 지정된 대상 브랜치와 병합합니다MyDemoRepo.

```
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
```

출력:

```
{
    "commitId": "4f178133EXAMPLE",
    "treeId": "389765daEXAMPLE"
}
```

자세한 내용은 AWS CodeCommit 사용 설명서의 브랜치 비교 및 병합을 참조하세요.

- 자세한 API 내용은 명령 참조MergeBranchesBySquash의 섹션을 참조하세요. AWS CLI

### merge-branches-by-three-way

다음 코드 예시에서는 merge-branches-by-three-way을 사용하는 방법을 보여 줍니다.

AWS CLI

3방향 병합 전략을 사용하여 두 브랜치를 병합하려면

다음 merge-branches-by-three-way 예제에서는 지정된 소스 브랜치를 라는 리포지토리의 지정된 대상 브랜치와 병합합니다MyDemoRepo.

```
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
```

출력:

```
{
    "commitId": "4f178133EXAMPLE",
    "treeId": "389765daEXAMPLE"
}
```

자세한 내용은 AWS CodeCommit 사용 설명서의 브랜치 비교 및 병합을 참조하세요.

- 자세한 API 내용은 명령 참조MergeBranchesByThreeWay의 섹션을 참조하세요. AWS CLI

### merge-pull-request-by-fast-forward

다음 코드 예시에서는 merge-pull-request-by-fast-forward을 사용하는 방법을 보여 줍니다.

AWS CLI

풀 요청을 병합하고 닫으려면

이 예제는 라는 리포지토리에서 '47'의 ID와 '99132ab0EXAMPLE'의 소스 커밋 ID로 풀 요청을 병합하고 닫는 방법을 보여줍니다MyDemoRepo.

```
aws codecommit merge-pull-request-by-fast-forward \
    --pull-request-id 47 \
    --source-commit-id 99132ab0EXAMPLE \
    --repository-name MyDemoRepo
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS CodeCommit 사용 설명서의 풀 요청 병합을 참조하세요.

- 자세한 API 내용은 명령 참조MergePullRequestByFastForward의 섹션을 참조하세요. AWS CLI

### `merge-pull-request-by-squash`

다음 코드 예시에서는 `merge-pull-request-by-squash`을 사용하는 방법을 보여 줍니다.

AWS CLI

스쿼시 병합 전략을 사용하여 풀 요청을 병합하려면

다음 `merge-pull-request-by-squash` 예제에서는 라는 리포지토리에서 ACCEPT_SOURCE의 충돌 해결 전략을 사용하여 지정된 풀 요청을 병합하고 닫습니다MyDemoRepo.

```
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"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS CodeCommit 사용 설명서의 풀 요청 병합을 참조하세요.

- 자세한 API 내용은 명령 참조MergePullRequestBySquash의 섹션을 참조하세요. AWS CLI

### `merge-pull-request-by-three-way`

다음 코드 예시에서는 `merge-pull-request-by-three-way`을 사용하는 방법을 보여 줍니다.

AWS CLI

3방향 병합 전략을 사용하여 풀 요청을 병합하려면

다음 merge-pull-request-by-three-way 예제에서는 라는 리포지토리에서 충돌 세부 정보 및 충돌 해결 전략에 대한 기본 옵션을 사용하여 지정된 풀 요청을 병합하고 닫습니다MyDemoRepo.

```
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"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS CodeCommit 사용 설명서의 풀 요청 병합을 참조하세요.

- 자세한 API 내용은 명령 참조MergePullRequestByThreeWay의 섹션을 참조하세요. AWS CLI

### override-pull-request-approval-rules

다음 코드 예시에서는 override-pull-request-approval-rules을 사용하는 방법을 보여 줍니다.

AWS CLI

풀 요청에 대한 승인 규칙 요구 사항을 재정의하려면

다음 override-pull-request-approval-rules 예제에서는 지정된 풀 요청에 대한 승인 규칙을 재정의합니다. 대신 재정의를 취소하려면 --override-status 파라미터 값을 로 설정합니다REVOKE.

```
aws codecommit override-pull-request-approval-rules \
    --pull-request-id 34  \
    --revision-id 927df8d8EXAMPLE \
    --override-status OVERRIDE
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS CodeCommit 사용 설명서의 풀 요청에 대한 승인 규칙 재정의를 참조하세요.

- 자세한 API 내용은 명령 참조OverridePullRequestApprovalRules의 섹션을 참조하세요. AWS CLI

### post-comment-for-compared-commit

다음 코드 예시에서는 post-comment-for-compared-commit을 사용하는 방법을 보여 줍니다.

AWS CLI

커밋에 대한 주석을 생성하려면

이 예제에서는 라는 리포지토리에서 두 커밋 간의 비교에서 cl_sample.js 파일에 변경 사항에 "Can you add a test case for this?" 대한 설명을 추가하는 방법을 보여줍니다MyDemoRepo.

```
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
```

출력:

```
{
    "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"
    }
}
```

- 자세한 API 내용은 명령 참조PostCommentForComparedCommit의 섹션을 참조하세요. AWS CLI

### post-comment-for-pull-request

다음 코드 예시에서는 post-comment-for-pull-request을 사용하는 방법을 보여 줍니다.

AWS CLI

풀 요청에 주석을 추가하려면

다음 post-comment-for-pull-request 예제에서는 '어디에서도 사용되지 않는 것처럼 보입니다. Can we remove them?"라는 주석을 쓴다고 가정하겠습니다. 라는 리포지토리에서 ID가 인 풀 요청의 ahs_count.py 파일 변경 47 시MyDemoRepo.

```
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
```

출력:

```
{
     "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"
}
```

- 자세한 API 내용은 명령 참조PostCommentForPullRequest의 섹션을 참조하세요. AWS CLI

### post-comment-reply

다음 코드 예시에서는 post-comment-reply을 사용하는 방법을 보여 줍니다.

AWS CLI

커밋 또는 풀 요청에 대한 주석에 회신하려면

이 예제에서는 시스템 생성 ID가 인 주석"Good catch. I'll remove them."에 대한 응답을 추가하는 방법을 보여줍니다abcd1234EXAMPLEb5678efgh.

```
aws codecommit post-comment-reply \
    --in-reply-to abcd1234EXAMPLEb5678efgh \
    --content "Good catch. I'll remove them." \
    --client-request-token 123Example
```

출력:

```
{
```

```
    "comment": {
        "authorArn": "arn:aws:iam::111111111111:user/Li_Juan",
        "clientRequestToken": "123Example",
        "commentId": "442b498bEXAMPLE5756813",
        "content": "Good catch. I'll remove them.",
        "creationDate": 1508369829.136,
        "deleted": false,
        "CommentId": "abcd1234EXAMPLEb5678efgh",
        "lastModifiedDate": 150836912.221,
        "callerReactions": [],
        "reactionCounts": []
    }
}
```

- 자세한 API 내용은 명령 참조PostCommentReply의 섹션을 참조하세요. AWS CLI

### put-comment-reaction

다음 코드 예시에서는 put-comment-reaction을 사용하는 방법을 보여 줍니다.

AWS CLI

이모티콘을 사용하여 커밋에 대한 주석에 회신하려면

다음 put-comment-reaction 예제에서는 ID가 이고 이모티콘 반응 값이 abcd1234EXAMPLEb5678efgh인 주석에 응답합니다:thumbsup:.

```
aws codecommit put-comment-reaction \
    --comment-id abcd1234EXAMPLEb5678efgh \
    --reaction-value :thumbsup:
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS CodeCommit 사용 설명서의 에서 커밋에 대한 설명을 AWS CodeCommit 참조하세요.

- 자세한 API 내용은 명령 참조PutCommentReaction의 섹션을 참조하세요. AWS CLI

### put-file

다음 코드 예시에서는 put-file을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리에 파일을 추가하려면

다음 put-file 예제에서는 'ExampleSolution.py'라는 파일을 'MyDemoRepo'라는 리포지토리에 'feature-randomizationfeature'라는 브랜치에 추가합니다. 브랜치의 가장 최근 커밋 ID는 '4c925148EXAMPLE'입니다.

```
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."
```

출력:

```
{
    "blobId": "2eb4af3bEXAMPLE",
    "commitId": "317f8570EXAMPLE",
    "treeId": "347a3408EXAMPLE"
}
```

- 자세한 API 내용은 명령 참조PutFile의 섹션을 참조하세요. AWS CLI

## put-repository-triggers

다음 코드 예시에서는 put-repository-triggers을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리에서 트리거를 추가하거나 업데이트하려면

이 예제에서는 라는 리포지토리에 대한 모든 트리거의 구조를 포함하는 이미 생성된 JSON 파일(여기서는 MyTriggers.json)을 사용하여 'MyFirstTrigger' 및 'MySecondTrigger'라는 트리거를 업데이트하는 방법을 보여줍니다 MyDemoRepo. 기존 트리거에 JSON 대해 를 가져오는 방법을 알아보려면 명령을 참조하세요 get-repository-triggers.

```
aws codecommit put-repository-triggers \
    --repository-name MyDemoRepo file://MyTriggers.json
```

MyTriggers.json의 콘텐츠:

```
{
    "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"
            ]
        }
    ]
}
```

출력:

```
{
    "configurationId": "6fa51cd8-35c1-EXAMPLE"
}
```

- 자세한 API 내용은 명령 참조PutRepositoryTriggers의 섹션을 참조하세요. AWS CLI

### tag-resource

다음 코드 예시에서는 tag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

기존 리포지토리에 AWS 태그를 추가하려면

다음 tag-resource 예제에서는 지정된 리포지토리에 두 개의 태그를 지정합니다.

```
aws codecommit tag-resource \
    --resource-arn arn:aws:codecommit:us-west-2:111111111111:MyDemoRepo \
    --tags Status=Secret,Team=Saanvi
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS CodeCommit 사용 설명서의 리포지토리에 태그 추가를 참조하세요.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### test-repository-triggers

다음 코드 예시에서는 test-repository-triggers을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리에서 트리거를 테스트하려면

이 예제에서는 라는 리 AWS CodeCommit 포지토리에서 'MyFirstTrigger'라는 트리거를 테스트하는 방법을 보여줍니다 MyDemoRepo. 이 예제에서는 리포지토리의 이벤트가 Amazon Simple Notification Service(AmazonSNS) 주제에서 알림을 트리거합니다.

명령:

```
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
```

출력:

```
{
  "successfulExecutions": [
```

```
        "MyFirstTrigger"
    ],
    "failedExecutions": []
}
```

- 자세한 API 내용은 명령 참조TestRepositoryTriggers의 섹션을 참조하세요. AWS CLI

### untag-resource

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리에서 AWS 태그를 제거하려면

다음 untag-resource 예제에서는 라는 리포지토리에서 지정된 키가 있는 태그를 제거합니다MyDemoRepo.

```
aws codecommit untag-resource \
    --resource-arn arn:aws:codecommit:us-west-2:111111111111:MyDemoRepo \
    --tag-keys Status
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS CodeCommit 사용 설명서의 리포지토리에서 태그 제거를 참조하세요.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-approval-rule-template-content

다음 코드 예시에서는 update-approval-rule-template-content을 사용하는 방법을 보여 줍니다.

AWS CLI

승인 규칙 템플릿의 내용을 업데이트하려면

다음 update-approval-rule-template-content 예제에서는 지정된 승인 규칙 템플릿의 내용을 변경하여 승인 풀을 의 역할을 맡는 사용자로 재정의합니다CodeCommitReview.

```
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/*\"]}]}"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS CodeCommit 사용 설명서의 승인 규칙 템플릿 관리를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateApprovalRuleTemplateContent의 섹션을 참조하세요. AWS CLI

### update-approval-rule-template-description

다음 코드 예시에서는 update-approval-rule-template-description을 사용하는 방법을 보여 줍니다.

AWS CLI

승인 규칙 템플릿에 대한 설명을 업데이트하려면

다음 update-approval-rule-template-description 예제에서는 지정된 승인 규칙 템플릿의 설명을 로 변경합니다Requires 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"
```

출력:

```
{
  "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"
  }
}
```

자세한 내용은 AWS CodeCommit 사용 설명서의 승인 규칙 템플릿 관리를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateApprovalRuleTemplateDescription의 섹션을 참조하세요. AWS CLI

### update-approval-rule-template-name

다음 코드 예시에서는 update-approval-rule-template-name을 사용하는 방법을 보여 줍니다.

AWS CLI

승인 규칙 템플릿의 이름을 업데이트하려면

다음 update-approval-rule-template-name 예제에서는 승인 규칙 템플릿의 이름을 에서 1-approver-rule-for-all-pull-requests`1-approver-rule로 변경합니다.

```
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
```

출력:

```
{
  "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"
  }
}
```

자세한 내용은 AWS CodeCommit 사용 설명서의 승인 규칙 템플릿 관리를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateApprovalRuleTemplateName의 섹션을 참조하세요. AWS CLI

### update-comment

다음 코드 예시에서는 update-comment을 사용하는 방법을 보여 줍니다.

AWS CLI

커밋에 대한 주석을 업데이트하려면

이 예제에서는 ID가 "Fixed as requested. I'll update the pull request."인 주석에 콘텐츠를 추가하는 방법을 보여줍니다442b498bEXAMPLE5756813.

```
aws codecommit update-comment \
    --comment-id 442b498bEXAMPLE5756813 \
    --content "Fixed as requested. I'll update the pull request."
```

출력:

```
{
    "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
            }
    }
}
```

- 자세한 API 내용은 명령 참조UpdateComment의 섹션을 참조하세요. AWS CLI

### update-default-branch

다음 코드 예시에서는 update-default-branch을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리의 기본 브랜치를 변경하려면

이 예제에서는 AWS CodeCommit 리포지토리의 기본 브랜치를 변경합니다. 이 명령은 오류가 있는 경우에만 출력을 생성합니다.

명령:

```
aws codecommit update-default-branch --repository-name MyDemoRepo --default-branch-
name MyNewBranch
```

출력:

```
None.
```

- 자세한 API 내용은 명령 참조UpdateDefaultBranch의 섹션을 참조하세요. AWS CLI

### update-pull-request-approval-rule-content

다음 코드 예시에서는 update-pull-request-approval-rule-content을 사용하는 방법을 보여 줍니다.

AWS CLI

풀 요청에 대한 승인 규칙을 편집하려면

다음 update-pull-request-approval-rule-content 예제에서는 123456789012 AWS 계정의 사용자를 포함하는 승인 풀에서 한 명의 IAM 사용자 승인을 요구하도록 승인 규칙을 지정했습니다.

```
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/*\"]}]}"
```

출력:

```
{
    "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",
    }
}
```

자세한 내용은 AWS CodeCommit 사용 설명서의 승인 규칙 편집 또는 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조UpdatePullRequestApprovalRuleContent의 섹션을 참조하세요. AWS CLI

### update-pull-request-approval-state

다음 코드 예시에서는 update-pull-request-approval-state을 사용하는 방법을 보여 줍니다.

AWS CLI

풀 요청에 대한 승인을 승인하거나 취소하려면

다음 update-pull-request-approval-state 예제에서는 ID가 27 이고 개정 ID가 인 풀 요청을 승인합니다9f29d167EXAMPLE. 대신 승인을 취소하려면 --approval-state 파라미터 값을로 설정합니다REVOKE.

```
aws codecommit update-pull-request-approval-state \
    --pull-request-id 27  \
    --revision-id 9f29d167EXAMPLE  \
    --approval-state "APPROVE"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS CodeCommit 사용 설명서의 풀 요청 검토를 참조하세요.

- 자세한 API 내용은 명령 참조UpdatePullRequestApprovalState의 섹션을 참조하세요. AWS CLI

### update-pull-request-description

다음 코드 예시에서는 update-pull-request-description을 사용하는 방법을 보여 줍니다.

AWS CLI

풀 요청에 대한 설명을 변경하려면

이 예제에서는 ID가 인 풀 요청에 대한 설명을 변경하는 방법을 보여줍니다47.

```
aws codecommit update-pull-request-description \
    --pull-request-id 47 \
    --description "Updated the pull request to remove unused global variable."
```

출력:

```
{
    "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"
    }
}
```

- 자세한 API 내용은 명령 참조UpdatePullRequestDescription의 섹션을 참조하세요. AWS CLI

### update-pull-request-status

다음 코드 예시에서는 update-pull-request-status을 사용하는 방법을 보여 줍니다.

AWS CLI

풀 요청의 상태를 변경하려면

이 예제에서는 ID가 인 풀 요청의 상태를 라는 AWS CodeCommit 리포지토리CLOSED의 42 상태로 변경하는 방법을 보여줍니다MyDemoRepo.

```
aws codecommit update-pull-request-status \
    --pull-request-id 42 \
    --pull-request-status CLOSED
```

출력:

```
{
```

```
    "pullRequest": {
        "approvalRules": [
            {
                "approvalRuleContent": "{\"Version\": \"2018-11-08\",\"Statements\":
 [{\"Type\": \"Approvers\",\"NumberOfApprovalsNeeded\": 2,\"ApprovalPoolMembers\":
 [\"arn:aws:sts::123456789012:assumed-role/CodeCommitReview/*\"]}]}",
                "approvalRuleId": "dd8b17fe-EXAMPLE",
                "approvalRuleName": "2-approvers-needed-for-this-change",
                "creationDate": 1571356106.936,
                "lastModifiedDate": 571356106.936,
                "lastModifiedUser": "arn:aws:iam::123456789012:user/Mary_Major",
                "ruleContentSha256": "4711b576EXAMPLE"
            }
        ],
        "authorArn": "arn:aws:iam::123456789012:user/Li_Juan",
        "clientRequestToken": "",
        "creationDate": 1508530823.165,
        "description": "Updated the pull request to remove unused global variable.",
        "lastActivityDate": 1508372423.12,
        "pullRequestId": "47",
        "pullRequestStatus": "CLOSED",
        "pullRequestTargets": [
            {
                "destinationCommit": "9f31c968EXAMPLE",
                "destinationReference": "refs/heads/main",
                "mergeMetadata": {
                    "isMerged": false,
                },
                "repositoryName": "MyDemoRepo",
                "sourceCommit": "99132ab0EXAMPLE",
                "sourceReference": "refs/heads/variables-branch"
            }
        ],
        "title": "Consolidation of global variables"
    }
}
```

- 자세한 API 내용은 명령 참조UpdatePullRequestStatus의 섹션을 참조하세요. AWS CLI

### update-pull-request-title

다음 코드 예시에서는 update-pull-request-title을 사용하는 방법을 보여 줍니다.

AWS CLI

풀 요청의 제목을 변경하려면

이 예제에서는 ID가 인 풀 요청의 제목을 변경하는 방법을 보여줍니다47.

```
aws codecommit update-pull-request-title \
    --pull-request-id 47 \
    --title "Consolidation of global variables - updated review"
```

출력:

```
{
    "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"
    }
}
```

- 자세한 API 내용은 명령 참조UpdatePullRequestTitle의 섹션을 참조하세요. AWS CLI

### update-repository-description

다음 코드 예시에서는 update-repository-description을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리에 대한 설명을 변경하려면

이 예제에서는 AWS CodeCommit 리포지토리에 대한 설명을 변경합니다. 이 명령은 오류가 있는 경우에만 출력을 생성합니다.

명령:

```
aws codecommit update-repository-description --repository-name MyDemoRepo --
repository-description "This description was changed"
```

출력:

```
None.
```

- 자세한 API 내용은 명령 참조UpdateRepositoryDescription의 섹션을 참조하세요. AWS CLI

### update-repository-name

다음 코드 예시에서는 update-repository-name을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리의 이름을 변경하려면

이 예제에서는 AWS CodeCommit 리포지토리의 이름을 변경합니다. 이 명령은 오류가 있는 경우에만 출력을 생성합니다. AWS CodeCommit 리포지토리의 이름을 변경하면 SSH 및 사용자가 리포지토리에 연결해야 HTTPS URLs 하는 가 변경됩니다. 사용자는 연결 설정을 업데이트할 때까지 이 리포지토리에 연결할 수 없습니다. 또한 리포지토리가 변경되므로 리포지토리 이름을 변경하면 이 리포지토리의 에 의존하는 모든 IAM 사용자 정책이 무효화ARN됩니다ARN.

명령:

```
aws codecommit update-repository-name --old-name MyDemoRepo --new-
name MyRenamedDemoRepo
```

출력:

```
None.
```

- 자세한 API 내용은 명령 참조UpdateRepositoryName의 섹션을 참조하세요. AWS CLI

## CodeDeploy 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 CodeDeploy.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### add-tags-to-on-premises-instances

다음 코드 예시에서는 add-tags-to-on-premises-instances을 사용하는 방법을 보여 줍니다.

AWS CLI

온프레미스 인스턴스에 태그를 추가하려면

다음 add-tags-to-on-premises-instances 예제는 동일한 온프레미스 인스턴스 태그의 AWS CodeDeploy 를 두 개의 온프레미스 인스턴스에 연결합니다. 에 온프레미스 인스턴스를 등록하지 않습니다 AWS CodeDeploy.

```
aws deploy add-tags-to-on-premises-instances \
    --instance-names AssetTag12010298EX AssetTag23121309EX \
    --tags Key=Name,Value=CodeDeployDemo-OnPrem
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조AddTagsToOnPremisesInstances의 섹션을 참조하세요. AWS CLI

### batch-get-application-revisions

다음 코드 예시에서는 batch-get-application-revisions을 사용하는 방법을 보여 줍니다.

AWS CLI

애플리케이션 개정에 대한 정보를 검색하려면

다음 batch-get-application-revisions 예제에서는 GitHub 리포지토리에 저장된 지정된 개정에 대한 정보를 검색합니다.

```
aws deploy batch-get-application-revisions \
    --application-name my-codedeploy-application \
    --revisions "[{\"gitHubLocation\": {\"commitId\":
 \"fa85936EXAMPLEa31736c051f10d77297EXAMPLE\",\"repository\": \"my-github-token/my-
repository\"},\"revisionType\": \"GitHub\"}]"
```

출력:

```
{
    "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": ""
}
```

자세한 내용은 참조BatchGetApplicationRevisions의 섹션을 참조하세요. AWS CodeDeploy API

- 자세한 API 내용은 명령 참조BatchGetApplicationRevisions의 섹션을 참조하세요. AWS CLI

### `batch-get-applications`

다음 코드 예시에서는 `batch-get-applications`을 사용하는 방법을 보여 줍니다.

AWS CLI

여러 애플리케이션에 대한 정보를 가져오려면

다음 `batch-get-applications` 예제에서는 사용자 AWS 계정과 연결된 여러 애플리케이션에 대한 정보를 표시합니다.

```
aws deploy batch-get-applications --application-names WordPress_App MyOther_App
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조BatchGetApplications의 섹션을 참조하세요. AWS CLI

### batch-get-deployment-groups

다음 코드 예시에서는 batch-get-deployment-groups을 사용하는 방법을 보여 줍니다.

AWS CLI

하나 이상의 배포 그룹에 대한 정보를 검색하려면

다음 batch-get-deployment-groups 예제에서는 지정된 CodeDeploy 애플리케이션과 연결된 두 배포 그룹에 대한 정보를 검색합니다.

```
aws deploy batch-get-deployment-groups \
    --application-name my-codedeploy-application \
    --deployment-group-names "[\"my-deployment-group-1\",\"my-deployment-group-2\"]"
```

출력:

```
{
    "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": ""
}
```

자세한 내용은 참조BatchGetDeploymentGroups의 섹션을 참조하세요. AWS CodeDeploy API

- 자세한 API 내용은 명령 참조BatchGetDeploymentGroups의 섹션을 참조하세요. AWS CLI

### batch-get-deployment-targets

다음 코드 예시에서는 batch-get-deployment-targets을 사용하는 방법을 보여 줍니다.

AWS CLI

배포와 연결된 대상을 검색하려면

다음 batch-get-deployment-targets 예제에서는 지정된 배포와 연결된 대상 중 하나에 대한 정보를 반환합니다.

```
aws deploy batch-get-deployment-targets \
    --deployment-id "d-1A2B3C4D5" \
    --target-ids "i-01a2b3c4d5e6f1111"
```

출력:

```
{
    "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"
            }
        }
    ]
}
```

자세한 내용은 참조BatchGetDeploymentTargets의 섹션을 참조하세요. AWS CodeDeploy API

- 자세한 API 내용은 명령 참조BatchGetDeploymentTargets의 섹션을 참조하세요. AWS CLI

### batch-get-deployments

다음 코드 예시에서는 batch-get-deployments을 사용하는 방법을 보여 줍니다.

AWS CLI

여러 배포에 대한 정보를 가져오려면

다음 batch-get-deployments 예제에서는 사용자 AWS 계정과 연결된 여러 배포에 대한 정보를 보여줍니다.

```
aws deploy batch-get-deployments --deployment-ids d-A1B2C3111 d-A1B2C3222
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조BatchGetDeployments의 섹션을 참조하세요. AWS CLI

### batch-get-on-premises-instances

다음 코드 예시에서는 batch-get-on-premises-instances을 사용하는 방법을 보여 줍니다.

AWS CLI

하나 이상의 온프레미스 인스턴스에 대한 정보를 가져오려면

다음 batch-get-on-premises-instances 예제에서는 두 개의 온프레미스 인스턴스에 대한 정보를 가져옵니다.

```
aws deploy batch-get-on-premises-instances --instance-
names AssetTag12010298EX AssetTag23121309EX
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조BatchGetOnPremisesInstances의 섹션을 참조하세요. AWS CLI

### `continue-deployment`

다음 코드 예시에서는 `continue-deployment`을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 대기 시간이 경과할 때까지 기다리지 않고 트래픽 경로 변경을 시작합니다.

다음 `continue-deployment` 예제에서는 트래픽을 대체 환경의 인스턴스로 전환하기 시작할 준비가 된 원래 환경의 인스턴스에서 트래픽 경로를 변경하기 시작합니다.

```
aws deploy continue-deployment \
    --deployment-id "d-A1B2C3111" \
    --deployment-wait-type "READY_WAIT"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 참조ContinueDeployment의 섹션을 참조하세요. AWS CodeDeploy API

- 자세한 API 내용은 명령 참조ContinueDeployment의 섹션을 참조하세요. AWS CLI

### `create-application`

다음 코드 예시에서는 `create-application`을 사용하는 방법을 보여 줍니다.

AWS CLI

애플리케이션을 생성하려면

다음 create-application 예제에서는 애플리케이션을 생성하고 사용자 AWS 계정과 연결합니다.

```
aws deploy create-application --application-name MyOther_App
```

출력:

```
{
    "applicationId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
}
```

- 자세한 API 내용은 명령 참조CreateApplication의 섹션을 참조하세요. AWS CLI

### create-deployment-config

다음 코드 예시에서는 create-deployment-config을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 배포 구성을 생성하려면

다음 create-deployment-config 예제에서는 사용자 지정 배포 구성을 생성하고 사용자 AWS 계정과 연결합니다.

```
aws deploy create-deployment-config \
    --deployment-config-name ThreeQuartersHealthy \
    --minimum-healthy-hosts type=FLEET_PERCENT,value=75
```

출력:

```
{
    "deploymentConfigId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
}
```

- 자세한 API 내용은 명령 참조CreateDeploymentConfig의 섹션을 참조하세요. AWS CLI

### create-deployment-group

다음 코드 예시에서는 create-deployment-group을 사용하는 방법을 보여 줍니다.

AWS CLI

배포 그룹을 생성하려면

다음 create-deployment-group 예제에서는 배포 그룹을 생성하고 이를 지정된 애플리케이션 및 사용자 AWS 계정과 연결합니다.

```
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
```

출력:

```
{
    "deploymentGroupId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
}
```

- 자세한 API 내용은 명령 참조CreateDeploymentGroup의 섹션을 참조하세요. AWS CLI

### create-deployment

다음 코드 예시에서는 create-deployment을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: EC2/온프레미스 컴퓨팅 플랫폼을 사용하여 CodeDeploy 배포 생성

다음 create-deployment 예제에서는 배포를 생성하고 사용자 AWS 계정과 연결합니다.

```
aws deploy create-deployment \
    --application-name WordPress_App \
    --deployment-config-name CodeDeployDefault.OneAtATime \
    --deployment-group-name WordPress_DG \
    --description "My demo deployment" \
```

```
    --s3-
location bucket=CodeDeployDemoBucket,bundleType=zip,eTag=dd56cfdEXAMPLE8e768f9d77fEXAMPLE,ke
```

출력:

```
{
    "deploymentId": "d-A1B2C3111"
}
```

예제 2: Amazon ECS 컴퓨팅 플랫폼을 사용하여 CodeDeploy 배포 생성

다음 `create-deployment` 예제에서는 다음 두 파일을 사용하여 Amazon ECS 서비스를 배포합니다.

`create-deployment.json` 파일의 콘텐츠:

```
{
    "applicationName": "ecs-deployment",
    "deploymentGroupName": "ecs-deployment-dg",
    "revision": {
        "revisionType": "S3",
        "s3Location": {
            "bucket": "ecs-deployment-bucket",
            "key": "appspec.yaml",
            "bundleType": "YAML"
        }
    }
}
```

그러면 이 파일은 라는 S3 버킷`appspec.yaml`에서 다음 파일을 검색합니다`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"
```

명령:

```
aws deploy create-deployment \
    --cli-input-json file://create-deployment.json \
    --region us-east-1
```

출력:

```
{
    "deploymentId": "d-1234ABCDE"
}
```

자세한 내용은 참조CreateDeployment의 섹션을 참조하세요. AWS CodeDeploy API

- 자세한 API 내용은 명령 참조CreateDeployment의 섹션을 참조하세요. AWS CLI

### delete-application

다음 코드 예시에서는 delete-application을 사용하는 방법을 보여 줍니다.

AWS CLI

애플리케이션 삭제

다음 delete-application 예제에서는 사용자 AWS 계정과 연결된 지정된 애플리케이션을 삭제합니다.

```
aws deploy delete-application --application-name WordPress_App
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeleteApplication의 섹션을 참조하세요. AWS CLI

### delete-deployment-config

다음 코드 예시에서는 delete-deployment-config을 사용하는 방법을 보여 줍니다.

AWS CLI

배포 구성을 삭제하려면

다음 delete-deployment-config 예제에서는 사용자 AWS 계정과 연결된 사용자 지정 배포 구성을 삭제합니다.

```
aws deploy delete-deployment-config --deployment-config-name ThreeQuartersHealthy
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeleteDeploymentConfig의 섹션을 참조하세요. AWS CLI

### delete-deployment-group

다음 코드 예시에서는 delete-deployment-group을 사용하는 방법을 보여 줍니다.

AWS CLI

배포 그룹을 삭제하려면

다음 delete-deployment-group 예제에서는 지정된 애플리케이션과 연결된 배포 그룹을 삭제합니다.

```
aws deploy delete-deployment-group \
    --application-name WordPress_App \
    --deployment-group-name WordPress_DG
```

출력:

```
{
    "hooksNotCleanedUp": []
}
```

- 자세한 API 내용은 명령 참조DeleteDeploymentGroup의 섹션을 참조하세요. AWS CLI

### delete-git-hub-account-token

다음 코드 예시에서는 delete-git-hub-account-token을 사용하는 방법을 보여 줍니다.

AWS CLI

GitHub 계정 연결을 삭제하려면

다음 delete-git-hub-account-token 예제에서는 지정된 GitHub 계정의 연결을 삭제합니다.

```
aws deploy delete-git-hub-account-token --token-name my-github-account
```

출력:

```
{
    "tokenName": "my-github-account"
}
```

자세한 내용은 참조DeleteGitHubAccountToken의 섹션을 참조하세요. AWS CodeDeploy API

- 자세한 API 내용은 명령 참조DeleteGitHubAccountToken의 섹션을 참조하세요. AWS CLI

### deregister-on-premises-instance

다음 코드 예시에서는 deregister-on-premises-instance을 사용하는 방법을 보여 줍니다.

AWS CLI

온프레미스 인스턴스 등록을 취소하려면

다음 deregister-on-premises-instance 예제에서는 에 온프레미스 인스턴스를 등록 취소하지만 AWS CodeDeploy인스턴스와 연결된 IAM 사용자를 삭제하지 않으며 인스턴스에서 온프레미스 인스턴스 태그 AWS CodeDeploy 의 연결을 해제하지도 않습니다. 또한 인스턴스에서 AWS CodeDeploy 에이전트를 제거하거나 인스턴스에서 온프레미스 구성 파일을 제거하지 않습니다.

```
aws deploy deregister-on-premises-instance --instance-name AssetTag12010298EX
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeregisterOnPremisesInstance의 섹션을 참조하세요. AWS CLI

### deregister

다음 코드 예시에서는 deregister을 사용하는 방법을 보여 줍니다.

AWS CLI

온프레미스 인스턴스 등록을 취소하려면

다음 deregister 예제에서는 를 사용하여 온프레미스 인스턴스의 등록을 취소합니다 AWS CodeDeploy. 인스턴스와 연결된 IAM 사용자는 삭제되지 않습니다. 인스턴스에서 온프레미스 태그

AWS CodeDeploy 의 연결을 해제합니다. 인스턴스에서 AWS CodeDeploy 에이전트를 제거하거나 인스턴스에서 온프레미스 구성 파일을 제거하지는 않습니다.

```
aws deploy deregister \
    --instance-name AssetTag12010298EX \
    --no-delete-iam-user \
    --region us-west-2
```

출력:

```
Retrieving on-premises instance information... DONE
IamUserArn: arn:aws:iam::80398EXAMPLE:user/AWS/CodeDeploy/AssetTag12010298EX
Tags: Key=Name,Value=CodeDeployDemo-OnPrem
Removing tags from the on-premises instance... DONE
Deregistering the on-premises instance... DONE
Run the following command on the on-premises instance to uninstall the codedeploy-
agent:
aws deploy uninstall
```

- API 자세한 내용은 AWS CLI 명령 참조의 등록 취소를 참조하세요.

### get-application-revision

다음 코드 예시에서는 get-application-revision을 사용하는 방법을 보여 줍니다.

AWS CLI

애플리케이션 개정에 대한 정보를 가져오려면

다음 get-application-revision 예제에서는 지정된 애플리케이션과 연결된 애플리케이션 개정에 대한 정보를 보여줍니다.

```
aws deploy get-application-revision \
    --application-name WordPress_App \
    --s3-
location bucket=CodeDeployDemoBucket,bundleType=zip,eTag=dd56cfdEXAMPLE8e768f9d77fEXAMPLE,ke
```

출력:

```
{
    "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"
        }
    }
}
```

- 자세한 API 내용은 명령 참조GetApplicationRevision의 섹션을 참조하세요. AWS CLI

### get-application

다음 코드 예시에서는 get-application을 사용하는 방법을 보여 줍니다.

AWS CLI

애플리케이션에 대한 정보를 가져오려면

다음 get-application 예제에서는 사용자 AWS 계정과 연결된 애플리케이션에 대한 정보를 표시합니다.

```
aws deploy get-application --application-name WordPress_App
```

출력:

```
{
    "application": {
        "applicationName": "WordPress_App",
        "applicationId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
        "createTime": 1407878168.078,
        "linkedToGitHub": false
```

```
    }
}
```

- 자세한 API 내용은 명령 참조GetApplication의 섹션을 참조하세요. AWS CLI

### get-deployment-config

다음 코드 예시에서는 get-deployment-config을 사용하는 방법을 보여 줍니다.

AWS CLI

배포 구성에 대한 정보를 가져오려면

다음 get-deployment-config 예제에서는 사용자 AWS 계정과 연결된 배포 구성에 대한 정보를 보여줍니다.

```
aws deploy get-deployment-config --deployment-config-name ThreeQuartersHealthy
```

출력:

```
{
    "deploymentConfigInfo": {
        "deploymentConfigId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
        "minimumHealthyHosts": {
            "type": "FLEET_PERCENT",
            "value": 75
        },
        "createTime": 1411081164.379,
        "deploymentConfigName": "ThreeQuartersHealthy"
    }
}
```

- 자세한 API 내용은 명령 참조GetDeploymentConfig의 섹션을 참조하세요. AWS CLI

### get-deployment-group

다음 코드 예시에서는 get-deployment-group을 사용하는 방법을 보여 줍니다.

AWS CLI

배포 그룹에 대한 정보를 보려면

다음 get-deployment-group 예제에서는 지정된 애플리케이션과 연결된 배포 그룹에 대한 정보를 보여줍니다.

```
aws deploy get-deployment-group \
    --application-name WordPress_App \
    --deployment-group-name WordPress_DG
```

출력:

```
{
    "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"
    }
}
```

- 자세한 API 내용은 명령 참조GetDeploymentGroup의 섹션을 참조하세요. AWS CLI

### get-deployment-instance

다음 코드 예시에서는 get-deployment-instance을 사용하는 방법을 보여 줍니다.

AWS CLI

배포 인스턴스에 대한 정보를 가져오려면

다음 get-deployment-instance 예제에서는 지정된 배포와 연결된 배포 인스턴스에 대한 정보를 보여줍니다.

```
aws deploy get-deployment-instance --deployment-id d-QA4G4F9EX --instance-
id i-902e9fEX
```

출력:

```
{
    "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"
    }
}
```

- 자세한 API 내용은 명령 참조GetDeploymentInstance의 섹션을 참조하세요. AWS CLI

### get-deployment-target

다음 코드 예시에서는 get-deployment-target을 사용하는 방법을 보여 줍니다.

AWS CLI

배포 대상에 대한 정보를 반환하려면

다음 get-deployment-target 예제에서는 지정된 배포와 연결된 배포 대상에 대한 정보를 반환합니다.

```
aws deploy get-deployment-target \
    --deployment-id "d-A1B2C3111" \
    --target-id "i-a1b2c3d4e5f611111"
```

출력:

```
{
    "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"
        }
    }
```

```
}
```

자세한 내용은 참조GetDeploymentTarget의 섹션을 참조하세요. AWS CodeDeploy API

- 자세한 API 내용은 명령 참조GetDeploymentTarget의 섹션을 참조하세요. AWS CLI

### get-deployment

다음 코드 예시에서는 get-deployment을 사용하는 방법을 보여 줍니다.

AWS CLI

배포에 대한 정보를 가져오려면

다음 get-deployment 예제에서는 사용자 AWS 계정과 연결된 배포에 대한 정보를 보여줍니다.

```
aws deploy get-deployment --deployment-id d-A1B2C3123
```

출력:

```
{
    "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
    }
}
```

- 자세한 API 내용은 명령 참조GetDeployment의 섹션을 참조하세요. AWS CLI

### get-on-premises-instance

다음 코드 예시에서는 get-on-premises-instance을 사용하는 방법을 보여 줍니다.

AWS CLI

온프레미스 인스턴스에 대한 정보를 가져오려면

다음 get-on-premises-instance 예제에서는 지정된 온프레미스 인스턴스에 대한 정보를 검색합니다.

```
aws deploy get-on-premises-instance --instance-name AssetTag12010298EX
```

출력:

```
{
    "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"
    }
}
```

- 자세한 API 내용은 명령 참조GetOnPremisesInstance의 섹션을 참조하세요. AWS CLI

### `install`

다음 코드 예시에서는 `install`을 사용하는 방법을 보여 줍니다.

AWS CLI

온프레미스 인스턴스를 설치하려면

다음 `install` 예제에서는 인스턴스의 지정된 위치에서 AWS CodeDeploy 에이전트가 찾을 것으로 예상되는 인스턴스의 위치로 온프레미스 구성 파일을 복사합니다. 또한 인스턴스에 AWS CodeDeploy 에이전트를 설치합니다. 사용자를 생성IAM하거나, 온프레미스 인스턴스를 에 등록하거나 AWS CodeDeploy, 인스턴스에 AWS CodeDeploy 대해 의 온프레미스 인스턴스 태그를 연결하지 않습니다.

```
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
```

출력:

```
Creating the on-premises instance configuration file... DONE
Installing the AWS CodeDeploy Agent... DONE
```

- 자세한 API 내용은 AWS CLI 명령 참조의 설치를 참조하세요.

### `list-application-revisions`

다음 코드 예시에서는 `list-application-revisions`을 사용하는 방법을 보여 줍니다.

AWS CLI

애플리케이션 개정에 대한 정보를 가져오려면

다음 `list-application-revisions` 예제에서는 지정된 애플리케이션과 연결된 모든 애플리케이션 개정에 대한 정보를 표시합니다.

```
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
```

출력:

```
{
    "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"
            }
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListApplicationRevisions의 섹션을 참조하세요. AWS CLI

### `list-applications`

다음 코드 예시에서는 `list-applications`을 사용하는 방법을 보여 줍니다.

AWS CLI

애플리케이션에 대한 정보를 가져오려면

다음 `list-applications` 예제에서는 사용자 AWS 계정과 연결된 모든 애플리케이션에 대한 정보를 표시합니다.

```
aws deploy list-applications
```

출력:

```
{
    "applications": [
        "WordPress_App",
        "MyOther_App"
    ]
}
```

- 자세한 API 내용은 명령 참조ListApplications의 섹션을 참조하세요. AWS CLI

### `list-deployment-configs`

다음 코드 예시에서는 `list-deployment-configs`을 사용하는 방법을 보여 줍니다.

AWS CLI

배포 구성에 대한 정보를 가져오려면

다음 `list-deployment-configs` 예제에서는 사용자 AWS 계정과 연결된 모든 배포 구성에 대한 정보를 표시합니다.

```
aws deploy list-deployment-configs
```

출력:

```
{
    "deploymentConfigsList": [
        "ThreeQuartersHealthy",
        "CodeDeployDefault.AllAtOnce",
        "CodeDeployDefault.HalfAtATime",
        "CodeDeployDefault.OneAtATime"
    ]
}
```

- 자세한 API 내용은 명령 참조ListDeploymentConfigs의 섹션을 참조하세요. AWS CLI

### list-deployment-groups

다음 코드 예시에서는 list-deployment-groups을 사용하는 방법을 보여 줍니다.

AWS CLI

배포 그룹에 대한 정보를 가져오려면

다음 list-deployment-groups 예제에서는 지정된 애플리케이션과 연결된 모든 배포 그룹에 대한 정보를 표시합니다.

```
aws deploy list-deployment-groups --application-name WordPress_App
```

출력:

```
{
    "applicationName": "WordPress_App",
    "deploymentGroups": [
        "WordPress_DG",
        "WordPress_Beta_DG"
    ]
}
```

- 자세한 API 내용은 명령 참조ListDeploymentGroups의 섹션을 참조하세요. AWS CLI

### list-deployment-instances

다음 코드 예시에서는 list-deployment-instances을 사용하는 방법을 보여 줍니다.

AWS CLI

배포 인스턴스에 대한 정보를 가져오려면

다음 list-deployment-instances 예제에서는 지정된 배포와 연결된 모든 배포 인스턴스에 대한 정보를 표시합니다.

```
aws deploy list-deployment-instances \
    --deployment-id d-A1B2C3111 \
    --instance-status-filter Succeeded
```

출력:

```
{
    "instancesList": [
        "i-EXAMPLE11",
        "i-EXAMPLE22"
    ]
}
```

- 자세한 API 내용은 명령 참조ListDeploymentInstances의 섹션을 참조하세요. AWS CLI

### `list-deployment-targets`

다음 코드 예시에서는 `list-deployment-targets`을 사용하는 방법을 보여 줍니다.

AWS CLI

배포와 IDs 연결된 대상 목록을 검색하려면

다음 `list-deployment-targets` 예제에서는 상태가 “실패” 또는 “.”인 배포와 IDs 연결된 대상 목록을 검색합니다InProgress.

```
aws deploy list-deployment-targets \
    --deployment-id "d-A1B2C3111" \
    --target-filters "{\"TargetStatus\":[\"Failed\",\"InProgress\"]}"
```

출력:

```
{
    "targetIds": [
        "i-0f1558aaf90e5f1f9"
    ]
}
```

자세한 내용은 참조ListDeploymentTargets의 섹션을 참조하세요. AWS CodeDeploy API

- 자세한 API 내용은 명령 참조ListDeploymentTargets의 섹션을 참조하세요. AWS CLI

### `list-deployments`

다음 코드 예시에서는 `list-deployments`을 사용하는 방법을 보여 줍니다.

AWS CLI

배포에 대한 정보를 가져오려면

다음 `list-deployments` 예제에서는 지정된 애플리케이션 및 배포 그룹과 연결된 모든 배포에 대한 정보를 표시합니다.

```
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
```

출력:

```
{
    "deployments": [
        "d-EXAMPLE11",
        "d-EXAMPLE22",
        "d-EXAMPLE33"
    ]
}
```

- 자세한 API 내용은 명령 참조ListDeployments의 섹션을 참조하세요. AWS CLI

### `list-git-hub-account-token-names`

다음 코드 예시에서는 `list-git-hub-account-token-names`을 사용하는 방법을 보여 줍니다.

AWS CLI

GitHub 계정에 저장된 연결의 이름을 나열하려면

다음 `list-git-hub-account-token-names` 예제에서는 현재 AWS 사용자의 GitHub 계정에 저장된 연결의 이름을 나열합니다.

```
aws deploy list-git-hub-account-token-names
```

출력:

```
{
```

```
    "tokenNameList": [
        "my-first-token",
        "my-second-token",
        "my-third-token"
    ]
}
```

자세한 내용은 참조ListGitHubAccountTokenNames의 섹션을 참조하세요. AWS CodeDeploy API

- 자세한 API 내용은 명령 참조ListGitHubAccountTokenNames의 섹션을 참조하세요. AWS CLI

### `list-on-premises-instances`

다음 코드 예시에서는 `list-on-premises-instances`을 사용하는 방법을 보여 줍니다.

AWS CLI

하나 이상의 온프레미스 인스턴스에 대한 정보를 가져오려면

다음 `list-on-premises-instances` 예제에서는 에 등록되어 AWS CodeDeploy 있고 에 인스턴스 AWS CodeDeploy 와 연결된 지정된 온프레미스 인스턴스 태그가 있는 인스턴스에 대해 사용 가능한 온프레미스 인스턴스 이름 목록을 검색합니다.

```
aws deploy list-on-premises-instances \
    --registration-status Registered \
    --tag-filters Key=Name,Value=CodeDeployDemo-OnPrem,Type=KEY_AND_VALUE
```

출력:

```
{
    "instanceNames": [
        "AssetTag12010298EX"
    ]
}
```

- 자세한 API 내용은 명령 참조ListOnPremisesInstances의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 대한 태그를 나열하려면(애플리케이션)

다음 `list-tags-for-resource` 예제에서는 에 이름이 지정된 애플리케이션에 적용된 태그를 나열 testApp 합니다 CodeDeploy.

```
aws deploy list-tags-for-resource \
    --resource-arn arn:aws:codedeploy:us-west-2:111122223333:application:testApp
```

출력:

```
{
    "Tags": [
        {
            "Key": "Type",
            "Value": "testType"
        },
        {
            "Key": "Name",
            "Value": "testName"
        }
    ]
}
```

자세한 내용은 AWS CodeDeploy 사용 설명서의 에서 배포 그룹에 대한 인스턴스 태그 지정 CodeDeploy을 참조하세요.

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### push

다음 코드 예시에서는 push을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon S3에 AWS CodeDeploy 호환되는 애플리케이션 개정을 번들링하고 배포하려면

다음 push 예제에서는 Amazon S3에 애플리케이션 개정을 번들링 및 배포한 다음 애플리케이션 개정을 지정된 애플리케이션과 연결합니다.

```
aws deploy push \
```

```
    --application-name WordPress_App \
    --description "This is my deployment" \
    --ignore-hidden-files \
    --s3-location s3://CodeDeployDemoBucket/WordPressApp.zip \
    --source /tmp/MyLocalDeploymentFolder/
```

출력은 create-deployment 명령을 사용하여 업로드된 애플리케이션 개정을 사용하는 배포를 생성하는 방법을 설명합니다.

```
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
```

- 자세한 API 내용은 Push in AWS CLI Command 참조를 참조하세요.

### register-application-revision

다음 코드 예시에서는 register-application-revision을 사용하는 방법을 보여 줍니다.

AWS CLI

이미 업로드된 애플리케이션 개정에 대한 정보를 등록하려면

다음 register-application-revision 예제에서는 Amazon S3에 저장된 이미 업로드된 애플리케이션 개정에 대한 정보를 에 등록합니다 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
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조RegisterApplicationRevision의 섹션을 참조하세요. AWS CLI

### register-on-premises-instance

다음 코드 예시에서는 register-on-premises-instance을 사용하는 방법을 보여 줍니다.

AWS CLI

온프레미스 인스턴스를 등록하려면

다음 `register-on-premises-instance` 예제에서는 온프레미스 인스턴스를 에 등록합니다 AWS CodeDeploy. 지정된 IAM 사용자를 생성하지 않으며 AWS CodeDeploy 온프레미스 인스턴스 태그에서 등록된 인스턴스와 연결하지도 않습니다.

```
aws deploy register-on-premises-instance \
    --instance-name AssetTag12010298EX \
    --iam-user-arn arn:aws:iam::80398EXAMPLE:user/CodeDeployDemoUser-OnPrem
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조RegisterOnPremisesInstance의 섹션을 참조하세요. AWS CLI

### `register`

다음 코드 예시에서는 `register`을 사용하는 방법을 보여 줍니다.

AWS CLI

온프레미스 인스턴스를 등록하려면

다음 `register` 예제에서는 온프레미스 인스턴스를 에 등록하고 AWS CodeDeploy, AWS CodeDeploy 지정된 온프레미스 인스턴스 태그에 등록된 인스턴스와 연결하고, 인스턴스에 복사할 수 있는 온프레미스 구성 파일을 생성합니다. 사용자를 생성IAM하지 않으며 인스턴스에 AWS CodeDeploy 에이전트를 설치하지도 않습니다.

```
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
```

출력:

```
Registering the on-premises instance... DONE
Adding tags to the on-premises instance... DONE
Copy the on-premises configuration file named codedeploy.onpremises.yml to the on-
premises instance, and run the following command on the on-premises instance to
 install and configure the AWS CodeDeploy Agent:
```

```
aws deploy install --config-file codedeploy.onpremises.yml
```

- API 자세한 내용은 AWS CLI 명령 참조의 등록을 참조하세요.

### `remove-tags-from-on-premises-instances`

다음 코드 예시에서는 `remove-tags-from-on-premises-instances`을 사용하는 방법을 보여 줍니다.

AWS CLI

하나 이상의 온프레미스 인스턴스에서 태그를 제거하려면

다음 `remove-tags-from-on-premises-instances` 예제에서는 의 지정된 온프레미스 태그를 온프레미스 인스턴스 AWS CodeDeploy 에서 연결 해제합니다. 에서 온프레미스 인스턴스를 등록 취소하거나 AWS CodeDeploy인스턴스에서 AWS CodeDeploy 에이전트를 제거하거나 인스턴스에서 온프레미스 구성 파일을 제거하거나 인스턴스와 연결된 IAM 사용자를 삭제하지 않습니다.

```
aws deploy remove-tags-from-on-premises-instances \
    --instance-names AssetTag12010298EX AssetTag23121309EX \
    --tags Key=Name,Value=CodeDeployDemo-OnPrem
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조RemoveTagsFromOnPremisesInstances의 섹션을 참조하세요. AWS CLI

### `stop-deployment`

다음 코드 예시에서는 `stop-deployment`을 사용하는 방법을 보여 줍니다.

AWS CLI

배포 중지를 시도하려면

다음 `stop-deployment` 예제에서는 사용자 AWS 계정과 연결된 진행 중인 배포를 중지하려고 시도합니다.

aws 배포 중지-배포 --deployment-id d-A1B2C3111

출력:

```
{
    "status": "Succeeded",
    "statusMessage": "No more commands will be scheduled for execution in the
 deployment instances"
}
```

- 자세한 API 내용은 명령 참조StopDeployment의 섹션을 참조하세요. AWS CLI

### `tag-resource`

다음 코드 예시에서는 `tag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소어에 태그를 지정하려면(애플리케이션)

다음 `tag-resource` 예제에서는 키 이름 및 유형과 testName 값 및 testType 가 포함된 태그 두 개를 testApp 에 이름이 지정된 애플리케이션에 추가합니다 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
```

성공하면 이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS CodeDeploy 사용 설명서의 에서 배포 그룹에 대한 인스턴스 태그 지정 CodeDeploy을 참조하세요.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### `uninstall`

다음 코드 예시에서는 `uninstall`을 사용하는 방법을 보여 줍니다.

AWS CLI

온프레미스 인스턴스를 제거하려면

다음 `uninstall` 예제에서는 온프레미스 인스턴스에서 AWS CodeDeploy 에이전트를 제거하고 인스턴스에서 온프레미스 구성 파일을 제거합니다. 에서 인스턴스 등록을 취소하거나 AWS CodeDeploy, 인스턴스 AWS CodeDeploy 에서 의 온프레미스 인스턴스 태그를 연결 해제하거나, 인스턴스와 연결된 IAM 사용자를 삭제하지 않습니다.

```
aws deploy uninstall
```

이 명령은 출력을 생성하지 않습니다.

- API 자세한 내용은 AWS CLI 명령 참조의 제거를 참조하세요.

### untag-resource

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에서 태그를 제거하려면(애플리케이션)

다음 untag-resource 예제에서는 testApp 에 이름이 지정된 애플리케이션에서 키 이름 및 유형이 있는 태그 2개를 제거합니다 CodeDeploy.

```
aws deploy untag-resource \
    --resource-arn  arn:aws:codedeploy:us-west-2:111122223333:application:testApp \
    --tag-keys Name Type
```

성공하면 이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS CodeDeploy 사용 설명서의 에서 배포 그룹에 대한 인스턴스 태그 지정 CodeDeploy을 참조하세요.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-application

다음 코드 예시에서는 update-application을 사용하는 방법을 보여 줍니다.

AWS CLI

애플리케이션의 세부 정보를 변경하려면

다음 update-application 예제에서는 사용자 AWS 계정과 연결된 애플리케이션의 이름을 변경합니다.

```
aws deploy update-application \
    --application-name WordPress_App \
    --new-application-name My_WordPress_App
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조UpdateApplication의 섹션을 참조하세요. AWS CLI

### update-deployment-group

다음 코드 예시에서는 update-deployment-group을 사용하는 방법을 보여 줍니다.

AWS CLI

배포 그룹에 대한 정보를 변경하려면

다음 update-deployment-group 예제에서는 지정된 애플리케이션과 연결된 배포 그룹의 설정을 변경합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조UpdateDeploymentGroup의 섹션을 참조하세요. AWS CLI

## CodeGuru 를 사용한 검토자 예제 AWS CLI

다음 코드 예제에서는 CodeGuru 검토자 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### `associate-repository`

다음 코드 예시에서는 `associate-repository`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: Bitbucket 리포지토리 연결을 생성하려면

다음 `associate-repository` 예제에서는 기존 Bitbucket 리포지토리를 사용하여 리포지토리 연결을 생성합니다.

```
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 }'
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon CodeGuru Reviewer 사용 설명서의 Amazon Reviewer에서 Bitbucket 리포지토리 연결 생성을 참조하세요 CodeGuru .

예제 2: GitHub 엔터프라이즈 리포지토리 연결을 생성하려면

다음 `associate-repository` 예제에서는 기존 GitHub 엔터프라이즈 리포지토리를 사용하여 리포지토리 연결을 생성합니다.

```
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 }'
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Codeguru CodeGuru Reviewer 사용 설명서의 Amazon Reviewer에서 GitHub 엔터프라이즈 서버 리포지토리 연결 생성을 참조하세요.

예제 3: AWS CodeCommit 리포지토리 연결을 생성하려면

다음 `associate-repository` 예제에서는 기존 리포지토리를 사용하여 AWS CodeCommit 리포지토리 연결을 생성합니다.

```
aws codeguru-reviewer associate-repository \
    --repository CodeCommit={Name=mySampleRepo}
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon CodeGuru Reviewer 사용 설명서의 Amazon Reviewer에서 AWS CodeCommit 리포지토리 연결 생성을 참조하세요. CodeGuru

- 자세한 API 내용은 명령 참조AssociateRepository의 섹션을 참조하세요. AWS CLI

### `create-code-review`

다음 코드 예시에서는 `create-code-review`을 사용하는 방법을 보여 줍니다.

AWS CLI

코드 검토를 생성합니다.

다음은 이름이 인 AWS CodeCommit 리포지토리의 `mainline` 브랜치에 코드 검토를 `create-code-review` 생성합니다`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"}}}'
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon CodeGuru Reviewer 사용 설명서의 Amazon Reviewer에서 코드 검토 생성을 참조하세요.  CodeGuru

- 자세한 API 내용은 명령 참조CreateCodeReview의 섹션을 참조하세요. AWS CLI

### `describe-code-review`

다음 코드 예시에서는 `describe-code-review`을 사용하는 방법을 보여 줍니다.

AWS CLI

코드 검토에 대한 세부 정보를 나열합니다.

다음은 "라는 AWS CodeCommit 리포지토리의 'mainlinemy-repo-name' 브랜치에 있는 코드 검토에 대한 정보를 `describe-code-review` 나열합니다.

```
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
```

출력

```
{
        "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"
        }
    }
```

자세한 내용은 Amazon CodeGuru Reviewer 사용 설명서의 코드 검토 세부 정보 보기를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeCodeReview의 섹션을 참조하세요. AWS CLI

### `describe-recommendation-feedback`

다음 코드 예시에서는 `describe-recommendation-feedback`을 사용하는 방법을 보여 줍니다.

AWS CLI

권장 사항에 대한 피드백 정보를 보려면

다음은 권장 사항에 대한 피드백에 대한 정보를 describe-recommendation-feedback 표시합니다. 이 권장 사항에는 한 가지 ThumbsUp 반응이 있습니다.

```
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
```

출력:

```
{
    "RecommendationFeedback": {
        "CodeReviewArn": "arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111:code-
review:RepositoryAnalysis-my-repository-name-branch-abcdefgh12345678",
        "RecommendationId":
 "3be1b2e5d7ef6e298a06499379ee290c9c596cf688fdcadb08285ddb0dd390eb",
        "Reactions": [
            "ThumbsUp"
        ],
        "UserId": "aws-user-id",
        "CreatedTimeStamp": 1618877070.313,
        "LastUpdatedTimeStamp": 1618877948.881
    }
}
```

자세한 내용은 Amazon CodeGuru Reviewer 사용 설명서의 권장 사항 보기 및 피드백 제공 및 4단계: 피드백 제공을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeRecommendationFeedback의 섹션을 참조하세요. AWS CLI

### describe-repository-association

다음 코드 예시에서는 describe-repository-association을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: GitHub 리포지토리 연결에 대한 정보를 반환하려면

다음 describe-repository-association 예제에서는 GitHub 엔터프라이즈 리포지토리를 사용하고 Associated 상태에 있는 리포지토리 연결에 대한 정보를 반환합니다.

```
aws codeguru-reviewer describe-repository-association \
    --association-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon CodeGuru Reviewer 사용 설명서의 Amazon Reviewer에서 GitHub 엔터프라이즈 서버 리포지토리 연결 생성을 참조하세요. CodeGuru

예제 2: 실패한 리포지토리 연결에 대한 정보를 반환하려면

다음 describe-repository-association 예제에서는 GitHub 엔터프라이즈 리포지토리를 사용하고 Failed 상태에 있는 리포지토리 연결에 대한 정보를 반환합니다.

```
aws codeguru-reviewer describe-repository-association \
    --association-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon CodeGuru Reviewer 사용 설명서의 Amazon Reviewer에서 GitHub 엔터프라이즈 서버 리포지토리 연결 생성을 참조하세요. CodeGuru

예제 3: 연결 해제 리포지토리 연결에 대한 정보를 반환하려면

다음 `describe-repository-association` 예제에서는 GitHub 엔터프라이즈 리포지토리를 사용하고 `Disassociating` 상태에 있는 리포지토리 연결에 대한 정보를 반환합니다.

```
aws codeguru-reviewer describe-repository-association \
    --association-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

출력:

```
{
    "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"
```

```
    }
}
```

자세한 내용은 Amazon CodeGuru Reviewer 사용 설명서의 Amazon Reviewer에서 GitHub 엔터프라이즈 서버 리포지토리 연결 생성을 참조하세요. CodeGuru

- 자세한 API 내용은 명령 참조DescribeRepositoryAssociation의 섹션을 참조하세요. AWS CLI

### `disassociate-repository`

다음 코드 예시에서는 `disassociate-repository`을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리 연결을 해제하려면

다음은 리포지토리를 사용하는 AWS CodeCommit 리포지토리 연결을 연결 `disassociate-repository` 해제합니다.

```
aws codeguru-reviewer disassociate-repository \
    --association-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon CodeGuru Reviewer 사용 설명서의 Reviewer에서 리포지토리 연결 해제를 참조하세요. CodeGuru

- 자세한 API 내용은 명령 참조DisassociateRepository의 섹션을 참조하세요. AWS CLI

### `list-code-reviews`

다음 코드 예시에서는 `list-code-reviews`을 사용하는 방법을 보여 줍니다.

AWS CLI

지난 90일 동안 AWS 계정에 생성된 코드 검토를 나열합니다.

다음 `list-code-reviews` 예제에서는 풀 요청을 사용하여 지난 90일 동안 생성된 코드 검토를 나열합니다.

```
aws codeguru-reviewer list-code-reviews \
    --type PullRequest
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon CodeGuru Reviewer 사용 설명서의 모든 코드 검토 보기를 참조하세요.

- 자세한 API 내용은 명령 참조ListCodeReviews의 섹션을 참조하세요. AWS CLI

### list-recommendation-feedback

다음 코드 예시에서는 list-recommendation-feedback을 사용하는 방법을 보여 줍니다.

AWS CLI

연결된 리포지토리의 추천에 대한 고객 추천 피드백을 나열하려면

다음은 코드 검토에 대한 모든 권장 사항에 대한 고객 피드백을 list-recommendation-feedback 나열합니다. 이 코드 검토에는 고객의 피드백 중 하나인 “ThumbsUp”가 있습니다.

```
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-abcdefgh12345678
```

출력:

```
{
    "RecommendationFeedbackSummaries": [
        {
            "RecommendationId":
 "3be1b2e5d7ef6e298a06499379ee290c9c596cf688fdcadb08285ddb0dd390eb",
            "Reactions": [
                "ThumbsUp"
            ],
            "UserId": "aws-user-id"
        }
    ]
}
```

자세한 내용은 Amazon CodeGuru Reviewer 사용 설명서의 4단계: 피드백 제공을 참조하세요.

- 자세한 API 내용은 명령 참조ListRecommendationFeedback의 섹션을 참조하세요. AWS CLI

### list-recommendations

다음 코드 예시에서는 list-recommendations을 사용하는 방법을 보여 줍니다.

AWS CLI

완료된 코드 검토에 대한 권장 사항을 나열하려면

다음 list-recommendations 예제에서는 완료된 코드 검토에 대한 권장 사항을 나열합니다. 이 코드 검토에는 한 가지 권장 사항이 있습니다.

```
aws codeguru-reviewer list-recommendations \
    --code-review-arn arn:aws:codeguru-reviewer:us-west-2:544120495673:code-review:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

출력:

```
{
    "RecommendationSummaries": [
        {
            "Description": "\n\n**Problem**  \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\n**Fix**  \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\n**More 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"
        }
    ]
}
```

자세한 내용은 Amazon CodeGuru Reviewer 사용 설명서의 4단계: 피드백 제공을 참조하세요.

- 자세한 API 내용은 명령 참조ListRecommendations의 섹션을 참조하세요. AWS CLI

### list-repository-associations

다음 코드 예시에서는 list-repository-associations을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 계정의 리포지토리 연결을 나열하려면

다음 list-repository-associations 예제에서는 계정의 리포지토리 연결 요약 객체 목록을 반환합니다. ProviderType, , 및 를 기준으로 반환된 목록을 필터링할 수 Name State있습니다Owner.

```
aws codeguru-reviewer list-repository-associations
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon CodeGuru Reviewer 사용 설명서의 Reviewer에서 모든 리포지토리 연결 보기를 참조하세요. CodeGuru

- 자세한 API 내용은 명령 참조ListRepositoryAssociations의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

연결된 리포지토리의 태그를 나열하려면

다음은 연결된 리포지토리의 태그를 `list-tags-for-resource` 나열합니다. 이 연결된 리포지토리에는 두 개의 태그가 있습니다.

```
aws codeguru-reviewer list-tags-for-resource \
    --resource-arn arn:aws:codeguru-reviewer:us-
west-2:123456789012:association:a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

출력:

```
{
    "Tags": {
        "Status": "Secret",
        "Team": "Saanvi"
    }
}
```

자세한 내용은 Amazon CodeGuru Reviewer 사용 설명서의 검토자 관련 리포지토리(AWS CLI)에 대한 태그 보기를 참조하세요. CodeGuru

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### put-recommendation-feedback

다음 코드 예시에서는 `put-recommendation-feedback`을 사용하는 방법을 보여 줍니다.

AWS CLI

코드 검토에 권장 사항을 추가하려면

다음은 코드 검토에 대한 ThumbsUp 권장 사항`put-recommendation-feedback`입니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon 검토자 사용 설명서의 4단계: 피드백 제공을 참조하세요. CodeGuru

- 자세한 API 내용은 명령 참조PutRecommendationFeedback의 섹션을 참조하세요. AWS CLI

### `tag-resource`

다음 코드 예시에서는 `tag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

연결된 리포지토리에 태그를 추가하려면

다음은 연결된 리포지토리에 두 개의 태그를 `tag-resource` 추가합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon CodeGuru Reviewer 사용 설명서의 검토자 관련 리포지토리에 태그 추가(AWS CLI) 및 CodeGuru 검토자 관련 리포지토리에 대한 태그 추가 또는 업데이트(AWS CLI)를 참조하세요. CodeGuru

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### `untag-resource`

다음 코드 예시에서는 `untag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

연결된 리포지토리의 태그를 해제하려면

다음은 연결된 리포지토리에서 'Secret' 및 'Team' 키가 있는 태그 2개를 `untag-resource` 제거합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon CodeGuru Reviewer 사용 설명서의 검토자 관련 리포지토리(AWS CLI)에서 태그 제거를 참조하세요. CodeGuru

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

## CodePipeline 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 CodePipeline.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



### 작업

#### `acknowledge-job`

다음 코드 예시에서는 `acknowledge-job`을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 작업에 대한 정보를 검색하려면

이 예제에서는 해당 작업이 있는 경우 해당 작업의 상태를 포함하여 지정된 작업에 대한 정보를 반환합니다. 이는 작업 작업자 및 사용자 지정 작업에만 사용됩니다. nonce 값과 작업 ID를 확인하려면 aws codepipeline 을 사용합니다 poll-for-jobs.

명령:

```
aws codepipeline acknowledge-job --job-id f4f4ff82-2d11-EXAMPLE --nonce 3
```

출력:

```
{
```

```
  "status": "InProgress"
}
```

- 자세한 API 내용은 명령 참조AcknowledgeJob의 섹션을 참조하세요. AWS CLI

### `create-custom-action-type`

다음 코드 예시에서는 `create-custom-action-type`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 작업을 생성하려면

이 예제에서는 사용자 지정 작업의 구조를 포함하는 이미 생성된 JSON 파일(여기서는 MyCustomAction.json)을 AWS CodePipeline 사용하기 위한 사용자 지정 작업을 생성합니다. 파일 구조를 포함하여 사용자 지정 작업을 생성하기 위한 요구 사항에 대한 자세한 내용은 AWS CodePipeline 사용 설명서를 참조하세요.

```
aws codepipeline create-custom-action-type --cli-input-json file://
MyCustomAction.json
```

JSON 파일 내용`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
    }
}
```

이 명령은 사용자 지정 작업의 구조를 반환합니다.

- 자세한 API 내용은 명령 참조CreateCustomActionType의 섹션을 참조하세요. AWS CLI

### `create-pipeline`

다음 코드 예시에서는 `create-pipeline`을 사용하는 방법을 보여 줍니다.

AWS CLI

파이프라인을 생성하려면

이 예제에서는 파이프라인의 구조를 포함하는 이미 생성된 JSON 파일(여기서는 MySecondPipeline.json)을 AWS CodePipeline 사용하여 에서 파이프라인을 생성합니다. 파일 구조를 포함하여 파이프라인 생성 요구 사항에 대한 자세한 내용은 AWS CodePipeline 사용 설명서를 참조하세요.

명령:

```
aws codepipeline create-pipeline --cli-input-json file://MySecondPipeline.json
```

JSON 파일 샘플 콘텐츠:

```
{
 "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
 }
}
```

출력:

```
This command returns the structure of the pipeline.
```

- 자세한 API 내용은 명령 참조CreatePipeline의 섹션을 참조하세요. AWS CLI

### `delete-custom-action-type`

다음 코드 예시에서는 `delete-custom-action-type`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 작업을 삭제하려면

이 예제에서는 삭제할 작업의 작업 유형, 공급자 이름 및 버전 번호가 포함된 이미 생성된 JSON 파일(여기서는 DeleteMyCustomAction.json)을 AWS CodePipeline 사용하여 에서 사용자 지정 작업을 삭제합니다. 명령을 사용하여 list-action-types 범주, 버전 및 공급자에 대한 올바른 값을 확인합니다.

명령:

```
aws codepipeline delete-custom-action-type --cli-input-json file://
DeleteMyCustomAction.json
```

JSON 파일 샘플 콘텐츠:

```
{
  "category": "Build",
```

```
  "version": "1",
  "provider": "MyJenkinsProviderName"
}
```

출력:

```
None.
```

- 자세한 API 내용은 명령 참조DeleteCustomActionType의 섹션을 참조하세요. AWS CLI

### delete-pipeline

다음 코드 예시에서는 delete-pipeline을 사용하는 방법을 보여 줍니다.

AWS CLI

파이프라인을 삭제하려면

이 예제에서는 MySecondPipeline 에서 라는 파이프라인을 삭제합니다 AWS CodePipeline. list-pipelines 명령을 사용하여 AWS 계정과 연결된 파이프라인 목록을 봅니다.

명령:

```
aws codepipeline delete-pipeline --name MySecondPipeline
```

출력:

```
None.
```

- 자세한 API 내용은 명령 참조DeletePipeline의 섹션을 참조하세요. AWS CLI

### delete-webhook

다음 코드 예시에서는 delete-webhook을 사용하는 방법을 보여 줍니다.

AWS CLI

웹후크를 삭제하려면

다음 delete-webhook 예제에서는 GitHub 버전 1 소스 작업에 대한 웹후크를 삭제합니다. deregister-webhook-with-third-party 명령을 사용하여 웹후크를 삭제하기 전에 등록을 취소해야 합니다.

```
aws codepipeline delete-webhook \
    --name my-webhook
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS CodePipeline 사용 설명서의 GitHub 소스에 대한 웹후크 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteWebhook의 섹션을 참조하세요. AWS CLI

### deregister-webhook-with-third-party

다음 코드 예시에서는 deregister-webhook-with-third-party을 사용하는 방법을 보여 줍니다.

AWS CLI

웹후크 등록을 취소하려면

다음 deregister-webhook-with-third-party 예제에서는 GitHub 버전 1 소스 작업에 대한 웹후크를 삭제합니다. Webhook를 삭제하려면 먼저 등록을 취소해야 합니다.

```
aws codepipeline deregister-webhook-with-third-party \
    --webhook-name my-webhook
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS CodePipeline 사용 설명서의 GitHub 소스에 대한 웹후크 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeregisterWebhookWithThirdParty의 섹션을 참조하세요. AWS CLI

### disable-stage-transition

다음 코드 예시에서는 disable-stage-transition을 사용하는 방법을 보여 줍니다.

AWS CLI

파이프라인에서 스테이지로의 전환을 비활성화하려면

이 예제에서는 에서 MyFirstPipeline 파이프라인의 베타 단계로의 전환을 비활성화합니다 AWS CodePipeline.

명령:

```
aws codepipeline disable-stage-transition --pipeline-name MyFirstPipeline --stage-name Beta  --transition-type Inbound
```

출력:

```
None.
```

- 자세한 API 내용은 명령 참조DisableStageTransition의 섹션을 참조하세요. AWS CLI

### enable-stage-transition

다음 코드 예시에서는 enable-stage-transition을 사용하는 방법을 보여 줍니다.

AWS CLI

파이프라인의 스테이지로 전환을 활성화하려면

이 예제에서는 에서 MyFirstPipeline 파이프라인의 베타 단계로 전환할 수 있습니다 AWS CodePipeline.

명령:

```
aws codepipeline enable-stage-transition --pipeline-name MyFirstPipeline --stage-name Beta  --transition-type Inbound
```

출력:

```
None.
```

- 자세한 API 내용은 명령 참조EnableStageTransition의 섹션을 참조하세요. AWS CLI

### get-job-details

다음 코드 예시에서는 get-job-details을 사용하는 방법을 보여 줍니다.

AWS CLI

작업의 세부 정보를 가져오려면

이 예제에서는 ID가 f4f4ff82-2d11-로 표시되는 작업에 대한 세부 정보를 반환합니다EXAMPLE. 이 명령은 사용자 지정 작업에만 사용됩니다. 이 명령이 호출되면 는 사용자 지정 작업에 필요한 경우 파이프라인의 아티팩트를 저장하는 데 사용되는 Amazon S3 버킷의 임시 자격 증명을 AWS CodePipeline 반환합니다. 이 명령은 작업에 대해 정의된 보안 암호 값이 정의된 경우에도 반환합니다.

명령:

```
aws codepipeline get-job-details --job-id f4f4ff82-2d11-EXAMPLE
```

출력:

```
{
 "jobDetails": {
  "accountId": "111111111111",
  "data": {
    "actionConfiguration": {
      "__type": "ActionConfiguration",
      "configuration": {
        "ProjectName": "MyJenkinsExampleTestProject"
      }
    },
    "actionTypeId": {
      "__type": "ActionTypeId",
      "category": "Test",
      "owner": "Custom",
      "provider": "MyJenkinsProviderName",
      "version": "1"
    },
    "artifactCredentials": {
      "__type": "AWSSessionCredentials",
      "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
      "secretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
      "sessionToken":
 "fICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwd
+a4GmWIWJ21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/
f0wYK8m9TrDHudUZg3qX4waLG5M43q7Wgc/
MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpEIbb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQ
+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0FkbFFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs
```

```
    },
    "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"
 }
}
```

- 자세한 API 내용은 명령 참조GetJobDetails의 섹션을 참조하세요. AWS CLI

### get-pipeline-state

다음 코드 예시에서는 get-pipeline-state을 사용하는 방법을 보여 줍니다.

AWS CLI

파이프라인 상태에 대한 정보를 가져오려면

이 예제는 라는 파이프라인의 최신 상태를 반환합니다 MyFirstPipeline.

명령:

```
aws codepipeline get-pipeline-state --name MyFirstPipeline
```

출력:

```
{
 "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
}
```

- 자세한 API 내용은 명령 참조GetPipelineState의 섹션을 참조하세요. AWS CLI

### get-pipeline

다음 코드 예시에서는 get-pipeline을 사용하는 방법을 보여 줍니다.

AWS CLI

파이프라인의 구조를 보려면

이 예제에서는 라는 파이프라인의 구조를 반환합니다 MyFirstPipeline.

명령:

```
aws codepipeline get-pipeline --name MyFirstPipeline
```

출력:

```
{
  "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
  }
}
```

- 자세한 API 내용은 명령 참조GetPipeline의 섹션을 참조하세요. AWS CLI

### `list-action-executions`

다음 코드 예시에서는 `list-action-executions`을 사용하는 방법을 보여 줍니다.

AWS CLI

작업 실행을 나열하려면

다음 `list-action-executions` 예제에서는 작업 실행 ID, 입력 아티팩트, 출력 아티팩트, 실행 결과 및 상태와 같은 파이프라인에 대한 작업 실행 세부 정보를 봅니다.

```
aws codepipeline list-action-executions \
    --pipeline-name myPipeline
```

출력:

```
{
    "actionExecutionDetails": [
        {
            "pipelineExecutionId": "EXAMPLE0-adfc-488e-bf4c-1111111720d3",
            "actionExecutionId": "EXAMPLE4-2ee8-4853-bd6a-111111158148",
            "pipelineVersion": 12,
            "stageName": "Deploy",
            "actionName": "Deploy",
            "startTime": 1598572628.6,
            "lastUpdateTime": 1598572661.255,
            "status": "Succeeded",
            "input": {
                "actionTypeId": {
                    "category": "Deploy",
                    "owner": "AWS",
                    "provider": "CodeDeploy",
                    "version": "1"
                },
                "configuration": {
                    "ApplicationName": "my-application",
                    "DeploymentGroupName": "my-deployment-group"
                },
                "resolvedConfiguration": {
                    "ApplicationName": "my-application",
                    "DeploymentGroupName": "my-deployment-group"
```

```
                },
                "region": "us-east-1",
                "inputArtifacts": [
                    {
                        "name": "SourceArtifact",
                        "s3location": {
                            "bucket": "artifact-bucket",
                            "key": "myPipeline/SourceArti/key"
                        }
                    }
                ],
                "namespace": "DeployVariables"
            },
            "output": {
                "outputArtifacts": [],
                "executionResult": {
                    "externalExecutionId": "d-EXAMPLEE5",
                    "externalExecutionSummary": "Deployment Succeeded",
                    "externalExecutionUrl": "https://myaddress.com"
                },
                "outputVariables": {}
            }
        },
        {
            "pipelineExecutionId": "EXAMPLE0-adfc-488e-bf4c-1111111720d3",
            "actionExecutionId": "EXAMPLE5-abb4-4192-9031-11111113a7b0",
            "pipelineVersion": 12,
            "stageName": "Source",
            "actionName": "Source",
            "startTime": 1598572624.387,
            "lastUpdateTime": 1598572628.16,
            "status": "Succeeded",
            "input": {
                "actionTypeId": {
                    "category": "Source",
                    "owner": "AWS",
                    "provider": "CodeCommit",
                    "version": "1"
                },
                "configuration": {
                    "BranchName": "production",
                    "PollForSourceChanges": "false",
                    "RepositoryName": "my-repo"
                },
```

```
                "resolvedConfiguration": {
                    "BranchName": "production",
                    "PollForSourceChanges": "false",
                    "RepositoryName": "my-repo"
                },
                "region": "us-east-1",
                "inputArtifacts": [],
                "namespace": "SourceVariables"
            },
            "output": {
                "outputArtifacts": [
                    {
                        "name": "SourceArtifact",
                        "s3location": {
                            "bucket": "my-bucket",
                            "key": "myPipeline/SourceArti/key"
                        }
                    }
                ],
                "executionResult": {
                    "externalExecutionId":
 "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"
                }
            }
        },
. . . .
```

자세한 내용은 AWS CodePipeline 사용 설명서의 작업 실행 보기(CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조ListActionExecutions의 섹션을 참조하세요. AWS CLI

### list-action-types

다음 코드 예시에서는 list-action-types을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 작업 유형을 보려면

자체적으로 사용되는 명령은 list-action-types AWS 계정에서 사용할 수 있는 모든 작업의 구조를 반환합니다. 이 예제에서는 --action-owner-filter 옵션을 사용하여 사용자 지정 작업만 반환합니다.

명령:

```
aws codepipeline list-action-types --action-owner-filter Custom
```

출력:

```
{
  "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}"
        }
    }
]
}
```

- 자세한 API 내용은 명령 참조ListActionTypes의 섹션을 참조하세요. AWS CLI

### `list-pipeline-executions`

다음 코드 예시에서는 `list-pipeline-executions`을 사용하는 방법을 보여 줍니다.

AWS CLI

파이프라인 실행 기록을 보려면

다음 `list-pipeline-executions` 예제는 AWS 계정의 파이프라인에 대한 파이프라인 실행 기록을 보여줍니다.

```
aws codepipeline list-pipeline-executions \
    --pipeline-name MyPipeline
```

출력:

```
{
    "pipelineExecutionSummaries": [
        {
            "lastUpdateTime": 1496380678.648,
            "pipelineExecutionId": "7cf7f7cb-3137-539g-j458-d7eu3EXAMPLE",
            "startTime": 1496380258.243,
            "status": "Succeeded"
        },
        {
            "lastUpdateTime": 1496591045.634,
            "pipelineExecutionId": "3137f7cb-8d494hj4-039j-d84l-d7eu3EXAMPLE",
            "startTime": 1496590401.222,
            "status": "Succeeded"
        },
        {
            "lastUpdateTime": 1496946071.6456,
            "pipelineExecutionId": "4992f7jf-7cf7-913k-k334-d7eu3EXAMPLE",
            "startTime": 1496945471.5645,
            "status": "Succeeded"
        }
    ]
}
```

자세한 내용은 AWS CodePipeline 사용 설명서의 실행 기록 보기를 참조하세요.

- 자세한 API 내용은 명령 참조ListPipelineExecutions의 섹션을 참조하세요. AWS CLI

### `list-pipelines`

다음 코드 예시에서는 `list-pipelines`을 사용하는 방법을 보여 줍니다.

AWS CLI

파이프라인 목록을 보려면

이 예제에서는 사용자 AWS 계정과 연결된 모든 AWS CodePipeline 파이프라인을 나열합니다.

명령:

```
aws codepipeline list-pipelines
```

출력:

```
{
  "pipelines": [
      {
          "updated": 1439504274.641,
          "version": 1,
          "name": "MyFirstPipeline",
          "created": 1439504274.641
      },
      {
          "updated": 1436461837.992,
          "version": 2,
          "name": "MySecondPipeline",
          "created": 1436460801.381
      }
      ]
}
```

- 자세한 API 내용은 명령 참조ListPipelines의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 지정된 파이프라인 리소스에 연결된 모든 태그 목록을 검색합니다.

```
aws codepipeline list-tags-for-resource \
    --resource-arn arn:aws:codepipeline:us-east-1:123456789012:MyPipeline
```

출력:

```
{
    "tags": {
        "Project": "ProjectA",
        "IscontainerBased": "true"
    }
}
```

자세한 내용은 AWS CodePipeline 사용 설명서의 파이프라인 태그 보기(CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### `list-webhooks`

다음 코드 예시에서는 `list-webhooks`을 사용하는 방법을 보여 줍니다.

AWS CLI

웹후크를 나열하려면

다음 `list-webhooks` 예제에서는 지정된 파이프라인 리소스에 연결된 모든 태그 목록을 검색합니다.

```
aws codepipeline list-webhooks \
    --endpoint-url "https://codepipeline.eu-central-1.amazonaws.com" \
    --region "eu-central-1"
```

출력:

```
{
    "webhooks": [
        {
            "url": "https://webhooks.domain.com/
trigger111111111EXAMPLE1111111111111111": {
                "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"
        }
    ]
}
```

자세한 내용은 AWS CodePipeline 사용 설명서의 계정에서 웹후크 나열을 참조하세요.

- 자세한 API 내용은 명령 참조ListWebhooks의 섹션을 참조하세요. AWS CLI

### poll-for-jobs

다음 코드 예시에서는 poll-for-jobs을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 작업을 보려면

이 예제에서는 작업 작업자가 조치를 취할 작업에 대한 정보를 반환합니다. 이 예제에서는 사전 정의된 JSON 파일(MyActionTypeInfo.json)을 사용하여 작업 작업자가 작업을 처리하는 작업 유형에 대한 정보를 제공합니다. 이 명령은 사용자 지정 작업에만 사용됩니다. 이 명령이 호출되면 는 파이프라인의 아티팩트를 저장하는 데 사용되는 Amazon S3 버킷의 임시 보안 인증을 AWS CodePipeline 반환합니다. 이 명령은 작업에 대해 정의된 보안 암호 값이 정의된 경우에도 반환합니다.

명령:

```
aws codepipeline poll-for-jobs --cli-input-json file://MyActionTypeInfo.json
```

JSON 파일 샘플 콘텐츠:

```
{
  "actionTypeId": {
    "category": "Test",
    "owner": "Custom",
    "provider": "MyJenkinsProviderName",
    "version": "1"
  },
  "maxBatchSize": 5,
  "queryParam": {
      "ProjectName": "MyJenkinsTestProject"
  }
}
```

출력:

```
{
 "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":
 "fICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwd
+a4GmWIWJ21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/
f0wYK8m9TrDHudUZg3qX4waLG5M43q7Wgc/
MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpEIbb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQ
+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0FkbFFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs
```

```
      },
      "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"
  }
 ]
}
```

- 자세한 API 내용은 명령 참조PollForJobs의 섹션을 참조하세요. AWS CLI

### put-webhook

다음 코드 예시에서는 put-webhook을 사용하는 방법을 보여 줍니다.

AWS CLI

웹후크를 생성하려면

다음 put-webhook 예제에서는 GitHub 버전 1 소스 작업에 대한 웹후크를 생성합니다. 웹후크를 생성한 후에는 register-webhook-with-third-party 명령을 사용하여 등록해야 합니다.

```
aws codepipeline put-webhook \
    --cli-input-json file://webhook_json.json \
    --region "eu-central-1"
```

webhook_json.json의 콘텐츠:

```
{
    "webhook": {
        "name": "my-webhook",
        "targetPipeline": "pipeline_name",
        "targetAction": "source_action_name",
        "filters": [
            {
                "jsonPath": "$.ref",
                "matchEquals": "refs/heads/{Branch}"
            }
        ],
        "authentication": "GITHUB_HMAC",
        "authenticationConfiguration": {
            "SecretToken": "secret"
        }
    }
}
```

출력:

```
{
    "webhook": {
        "url": "https://webhooks.domain.com/
trigger111111111EXAMPLE11111111111111111",
        "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"
        }
    ]
}
```

자세한 내용은 AWS CodePipeline 사용 설명서의 GitHub 소스에 대한 웹후크 생성을 참조하세요.

- 자세한 API 내용은 명령 참조PutWebhook의 섹션을 참조하세요. AWS CLI

### retry-stage-execution

다음 코드 예시에서는 `retry-stage-execution`을 사용하는 방법을 보여 줍니다.

AWS CLI

실패한 작업을 다시 시도하려면

다음 `retry-stage-execution` 예제에서는 실패한 작업이 있는 단계를 재시도합니다.

```
aws codepipeline retry-stage-execution \
    --pipeline-name MyPipeline \
    --stage-name Deploy \
    --pipeline-execution-id b59babff-5f34-EXAMPLE \
    --retry-mode FAILED_ACTIONS
```

출력:

```
{
    "pipelineExecutionId": "b59babff-5f34-EXAMPLE"
}
```

자세한 내용은 AWS CodePipeline 사용 설명서의 실패한 작업 재시도(CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조RetryStageExecution의 섹션을 참조하세요. AWS CLI

### start-pipeline-execution

다음 코드 예시에서는 start-pipeline-execution을 사용하는 방법을 보여 줍니다.

AWS CLI

파이프라인을 통해 최신 개정을 실행하려면

이 예제에서는 파이프라인의 소스 단계에 있는 최신 개정을 “MyFirstPipeline”라는 파이프라인을 통해 실행합니다.

명령:

```
aws codepipeline start-pipeline-execution --name MyFirstPipeline
```

출력:

```
{
  "pipelineExecutionId": "3137f7cb-7cf7-EXAMPLE"
}
```

- 자세한 API 내용은 명령 참조StartPipelineExecution의 섹션을 참조하세요. AWS CLI

### stop-pipeline-execution

다음 코드 예시에서는 stop-pipeline-execution을 사용하는 방법을 보여 줍니다.

AWS CLI

파이프라인 실행을 중지하려면

다음 stop-pipeline-execution 예제는 진행 중인 작업이 완료될 때까지 기다리는 것으로 기본 설정된 다음 파이프라인을 중지합니다. 실행이 이미 중지 상태인 경우 중지하고 대기하도록 선택할 수 없습니다. 이미 중지 상태인 실행을 중지하고 중단하도록 선택할 수 있습니다.

```
aws codepipeline stop-pipeline-execution \
    --pipeline-name MyFirstPipeline \
    --pipeline-execution-id d-EXAMPLE \
    --reason "Stopping pipeline after the build action is done"
```

이 명령은 출력을 반환하지 않습니다.

자세한 내용은 AWS CodePipeline 사용 설명서의 파이프라인 실행 중지(CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조StopPipelineExecution의 섹션을 참조하세요. AWS CLI

### tag-resource

다음 코드 예시에서는 tag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 태그를 지정하려면

다음 tag-resource 예제에서는 제공된 태그 세트를 파이프라인과 연결합니다. 태그를 추가하거나 편집하려면 이 명령을 사용합니다.

```
aws codepipeline tag-resource \
    --resource-arn arn:aws:codepipeline:us-east-1:123456789012:MyPipeline \
    --tags key=Project,value=ProjectA key=IscontainerBased,value=true
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS CodePipeline 사용 설명서의 파이프라인(CLI)에 태그 추가를 참조하세요.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### untag-resource

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

연결 리소스에서 AWS 태그를 제거하려면

다음 untag-resource 예제에서는 지정된 리소스에서 태그를 제거합니다.

```
aws codepipeline untag-resource \
    --resource-arn arn:aws:codepipeline:us-east-1:123456789012:MyPipeline \
    --tag-keys Project IscontainerBased
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS CodePipeline 사용 설명서의 파이프라인에서 태그 제거(CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-pipeline

다음 코드 예시에서는 update-pipeline을 사용하는 방법을 보여 줍니다.

AWS CLI

파이프라인의 구조를 업데이트하려면

이 예제에서는 update-pipeline 명령을 --cli-input-json argument와 함께 사용합니다. 이 예제에서는 사전 정의된 JSON 파일(MyFirstPipeline.json)을 사용하여 파이프라인의 구조를 업데이트합니다. 는 JSON 파일에 포함된 파이프라인 이름을 AWS CodePipeline 인식한 다음 파이프라인 구조의 수정된 필드의 변경 사항을 적용하여 파이프라인을 업데이트합니다.

사전 정의된 JSON 파일을 생성할 때 다음 지침을 사용합니다.

get-pipeline 명령을 사용하여 검색된 파이프라인 구조로 작업하는 경우 JSON 파일의 파이프라인 구조에서 메타데이터 섹션을 제거해야 합니다('메타데이터': { } 라인 및 '생성됨', '파이프라인'ARN, '업데이트됨' 필드).파이프라인 이름은 변경할 수 없습니다.

명령:

```
aws codepipeline update-pipeline --cli-input-json file://MyFirstPipeline.json
```

샘플 JSON 파일 내용:

```
{
 "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
 }
}
```

출력:

```
{
 "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
 }
}
```

- 자세한 API 내용은 명령 참조UpdatePipeline의 섹션을 참조하세요. AWS CLI

## AWS CodeStar 를 사용한 알림 예제 AWS CLI

다음 코드 예제에서는 AWS CodeStar 알림과 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `create-notification-rule`

다음 코드 예시에서는 `create-notification-rule`을 사용하는 방법을 보여 줍니다.

AWS CLI

알림 규칙을 생성하려면

다음 `create-notification-rule` 예제에서는 라는 JSON 파일을 사용하여 지정된 AWS 계정에 이름이 지정된 리포지토리MyDemoRepo에 MyNotificationRule 대한 라는 알림 규칙을 `rule.json` 생성합니다. 브랜치와 태그가 생성되면 FULL 세부 유형이 포함된 알림이 지정된 대상 Amazon SNS 주제로 전송됩니다.

```
aws codestar-notifications create-notification-rule \
    --cli-input-json file://rule.json
```

`rule.json`의 콘텐츠:

```
{
    "Name": "MyNotificationRule",
    "EventTypeIds": [
        "codecommit-repository-branches-and-tags-created"
    ],
    "Resource": "arn:aws:codecommit:us-east-1:123456789012:MyDemoRepo",
    "Targets": [
        {
            "TargetType": "SNS",
            "TargetAddress": "arn:aws:sns:us-
east-1:123456789012:MyNotificationTopic"
        }
    ],
    "Status": "ENABLED",
    "DetailType": "FULL"
```

```
}
```

출력:

```
{
    "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE"
}
```

자세한 내용은 AWS 개발자 도구 콘솔 사용 설명서의 알림 규칙 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateNotificationRule의 섹션을 참조하세요. AWS CLI

### delete-notification-rule

다음 코드 예시에서는 delete-notification-rule을 사용하는 방법을 보여 줍니다.

AWS CLI

알림 규칙을 삭제하려면

다음 delete-notification-rule 예제에서는 지정된 알림 규칙을 삭제합니다.

```
aws codestar-notifications delete-notification-rule \
    --arn arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE
```

출력:

```
{
    "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE"
}
```

자세한 내용은 AWS 개발자 도구 콘솔 사용 설명서의 알림 규칙 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteNotificationRule의 섹션을 참조하세요. AWS CLI

### delete-target

다음 코드 예시에서는 delete-target을 사용하는 방법을 보여 줍니다.

AWS CLI

알림 규칙 대상을 삭제하려면

다음 delete-target 예제에서는 지정된 대상을 대상으로 사용하도록 구성된 모든 알림 규칙에서 제거한 다음 대상을 삭제합니다.

```
aws codestar-notifications  delete-target \
    --target-address arn:aws:sns:us-east-1:123456789012:MyNotificationTopic \
    --force-unsubscribe-all
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS 개발자 도구 콘솔 사용 설명서의 알림 규칙 대상 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteTarget의 섹션을 참조하세요. AWS CLI

### describe-notification-rule

다음 코드 예시에서는 describe-notification-rule을 사용하는 방법을 보여 줍니다.

AWS CLI

알림 규칙의 세부 정보를 검색하려면

다음 describe-notification-rule 예제에서는 지정된 알림 규칙의 세부 정보를 검색합니다.

```
aws codestar-notifications describe-notification-rule \
    --arn arn:aws:codestar-notifications:us-west-2:123456789012:notificationrule/
dc82df7a-EXAMPLE
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS 개발자 도구 콘솔 사용 설명서의 알림 규칙 보기를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeNotificationRule의 섹션을 참조하세요. AWS CLI

### `list-event-types`

다음 코드 예시에서는 `list-event-types`을 사용하는 방법을 보여 줍니다.

AWS CLI

알림 규칙의 이벤트 유형 목록을 가져오려면

다음 `list-event-types` 예제에서는 CodeDeploy 애플리케이션에 사용할 수 있는 모든 알림 이벤트 유형의 필터링된 목록을 검색합니다. 대신 필터를 사용하지 않으면 명령은 모든 리소스 유형에 대해 모든 알림 이벤트 유형을 반환합니다.

```
aws codestar-notifications list-event-types \
    --filters Name=SERVICE_NAME,Value=CodeDeploy
```

출력:

```
{
```

```
    "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"
        }
    ]
}
```

자세한 내용은 AWS 개발자 도구 콘솔 사용 설명서의 알림 규칙 생성을 참조하세요.

- 자세한 API 내용은 명령 참조ListEventTypes의 섹션을 참조하세요. AWS CLI

### `list-notification-rules`

다음 코드 예시에서는 `list-notification-rules`을 사용하는 방법을 보여 줍니다.

AWS CLI

알림 규칙 목록을 검색하려면

다음 `list-notification-rules` 예제에서는 지정된 AWS 리전의 모든 알림 규칙 목록을 검색합니다.

```
aws codestar-notifications list-notification-rules --region us-east-1
```

출력:

```
{
```

```
    "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"
        }
    ]
}
```

자세한 내용은 AWS 개발자 도구 콘솔 사용 설명서의 알림 규칙 보기를 참조하세요.

- 자세한 API 내용은 명령 참조ListNotificationRules의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

알림 규칙에 연결된 태그 목록을 가져오려면

다음 `list-tags-for-resource` 예제에서는 지정된 알림 규칙에 연결된 모든 태그 목록을 검색합니다. 이 예제에서는 알림 규칙에 현재 연결된 태그가 없습니다.

```
aws codestar-notifications list-tags-for-resource \
    --arn arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
fe1efd35-EXAMPLE
```

출력:

```
{
    "Tags": {}
}
```

자세한 내용은 AWS 개발자 도구 콘솔 사용 설명서의 알림 규칙 생성을 참조하세요.

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### list-targets

다음 코드 예시에서는 list-targets을 사용하는 방법을 보여 줍니다.

AWS CLI

알림 규칙 대상 목록을 검색하려면

다음 list-targets 예제에서는 지정된 AWS 리전의 모든 알림 규칙 대상 목록을 검색합니다.

```
aws codestar-notifications list-targets \
    --region us-east-1
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS 개발자 도구 콘솔 사용 설명서의 알림 규칙 대상 보기를 참조하세요.

- 자세한 API 내용은 명령 참조ListTargets의 섹션을 참조하세요. AWS CLI

### subscribe

다음 코드 예시에서는 subscribe을 사용하는 방법을 보여 줍니다.

AWS CLI

알림 규칙에 대상을 추가하려면

다음 subscribe 예제에서는 지정된 알림 규칙의 대상으로 Amazon SNS 주제를 추가합니다.

```
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
```

출력:

```
{
    "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/dc82df7a-EXAMPLE"
}
```

자세한 내용은 AWS 개발자 도구 콘솔 사용 설명서의 알림 규칙의 대상으로 Amazon SNS 주제 추가 또는 제거를 참조하세요.

- API 자세한 내용은 AWS CLI 명령 참조의 구독을 참조하세요.

### tag-resource

다음 코드 예시에서는 tag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

알림 규칙에 태그를 추가하려면

다음 tag-resource 예제에서는 키 이름이 Team 이고 값이 인 태그를 Li_Juan 지정된 알림 규칙에 추가합니다.

```
aws codestar-notifications tag-resource \
    --arn arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/fe1efd35-EXAMPLE \
    --tags Team=Li_Juan
```

출력:

```
{
    "Tags": {
        "Team": "Li_Juan"
    }
```

```
}
```

자세한 내용은 AWS 개발자 도구 콘솔 사용 설명서의 알림 규칙 생성을 참조하세요.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### `unsubscribe`

다음 코드 예시에서는 unsubscribe을 사용하는 방법을 보여 줍니다.

AWS CLI

알림 규칙에서 대상을 제거하려면

다음 unsubscribe 예제에서는 지정된 알림 규칙에서 Amazon SNS 주제를 대상으로 제거합니다.

```
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
```

출력:

```
{
    "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE"
    "TargetAddress": "arn:aws:sns:us-east-1:123456789012:MyNotificationTopic"
}
```

자세한 내용은 AWS 개발자 도구 콘솔 사용 설명서의 알림 규칙의 대상으로 Amazon SNS 주제 추가 또는 제거를 참조하세요.

- API 자세한 내용은 AWS CLI 명령 참조의 구독 취소를 참조하세요.

### `untag-resource`

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

알림 규칙에서 태그를 제거하려면

다음 untag-resource 예제에서는 지정된 알림 규칙Team에서 키 이름이 인 태그를 제거합니다.

```
aws codestar-notifications untag-resource \
    --arn arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
fe1efd35-EXAMPLE \
    --tag-keys Team
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS 개발자 도구 콘솔 사용 설명서의 알림 규칙 편집을 참조하세요.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-notification-rule

다음 코드 예시에서는 update-notification-rule을 사용하는 방법을 보여 줍니다.

AWS CLI

알림 규칙을 업데이트하려면

다음 update-notification-rule 예제에서는 라는 JSON 파일을 123456789012 사용하여 AWS 계정에 MyNotificationRule 라는 알림 규칙을 업데이트합니다update.json.

```
aws codestar-notifications update-notification-rule \
    --cli-input-json file://update.json
```

update.json의 콘텐츠:

```
{
    "Name": "MyUpdatedNotificationRule",
    "EventTypeIds": [
        "codecommit-repository-branches-and-tags-created"
    ],
    "Resource": "arn:aws:codecommit:us-east-1:123456789012:MyDemoRepo",
    "Targets": [
        {
            "TargetType": "SNS",
            "TargetAddress": "arn:aws:sns:us-
east-1:123456789012:MyNotificationTopic"
        }
```

```
    ],
    "Status": "ENABLED",
    "DetailType": "FULL"
}
```

출력:

```
{
    "Arn": "arn:aws:codestar-notifications:us-east-1:123456789012:notificationrule/
dc82df7a-EXAMPLE"
}
```

자세한 내용은 AWS 개발자 도구 콘솔 사용 설명서의 알림 규칙 편집을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateNotificationRule의 섹션을 참조하세요. AWS CLI

## CodeConnections 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 CodeConnections.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `create-connection`

다음 코드 예시에서는 `create-connection`을 사용하는 방법을 보여 줍니다.

AWS CLI

연결을 생성하려면

다음 create-connection 예제에서는 타사 리포지토리에 대한 연결을 생성하는 방법을 보여줍니다. 이 예제에서는 타사 공급자가 Bitbucket인 연결을 생성합니다.

또는 를 AWS CLI 통해 생성된 연결 AWS CloudFormation 은 기본적으로 보류 중 상태입니다. CLI 또는 와의 연결을 생성한 후 콘솔을 AWS CloudFormation사용하여 연결을 편집하여 상태를 사용 가능하게 만듭니다.

```
aws codestar-connections create-connection \
    --provider-type Bitbucket \
    --connection-name MyConnection
```

출력:

```
{
    "ConnectionArn": "arn:aws:codestar-connections:us-
east-1:123456789012:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f"
}
```

자세한 내용은 개발자 도구 콘솔 사용 설명서의 연결 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateConnection의 섹션을 참조하세요. AWS CLI

### create-host

다음 코드 예시에서는 create-host을 사용하는 방법을 보여 줍니다.

AWS CLI

호스트를 생성하려면

다음 create-host 예제에서는 타사 공급자가 설치된 인프라의 엔드포인트를 나타내는 호스트를 생성하는 방법을 보여줍니다. 이 예제에서는 타사 설치 공급자가 GitHub Enterprise Server인 호스트를 생성합니다.

를 AWS CLI 통해 생성된 호스트는 기본적으로 보류 중 상태입니다. 를 사용하여 호스트를 생성한 후 콘솔 또는 를 CLI사용하여 호스트CLI의 상태를 사용 가능하게 설정하세요.

```
aws codestar-connections create-host \
    --name MyHost \
    --provider-type GitHubEnterpriseServer \
    --provider-endpoint "https://my-instance.dev"
```

출력:

```
{
    "HostArn": "arn:aws:codestar-connections:us-east-1:123456789012:host/My-
Host-28aef605"
}
```

자세한 내용은 개발자 도구 콘솔 사용 설명서의 호스트 생성(CLI)을 참조하세요.

- 자세한 API 내용은 명령 참조CreateHost의 섹션을 참조하세요. AWS CLI

### delete-connection

다음 코드 예시에서는 delete-connection을 사용하는 방법을 보여 줍니다.

AWS CLI

연결을 삭제하려면

다음 delete-connection 예제에서는 연결을 삭제하는 방법을 보여줍니다.

```
aws codestar-connections delete-connection \
    --connection-arn arn:aws:codestar-connections:us-west-2:123456789012:connection/
aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 개발자 도구 콘솔 사용 설명서의 연결 삭제(CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteConnection의 섹션을 참조하세요. AWS CLI

### delete-host

다음 코드 예시에서는 delete-host을 사용하는 방법을 보여 줍니다.

AWS CLI

호스트를 삭제하려면

다음 delete-host 예제에서는 호스트를 삭제하는 방법을 보여줍니다. 호스트를 삭제하려면 먼저 호스트와 연결된 모든 연결을 삭제해야 합니다.

```
aws codestar-connections delete-host \
```

```
    --host-arn "arn:aws:codestar-connections:us-east-1 :123456789012:host/My-
Host-28aef605"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 개발자 도구 콘솔 사용 설명서의 호스트 삭제(CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteHost의 섹션을 참조하세요. AWS CLI

### get-connection

다음 코드 예시에서는 get-connection을 사용하는 방법을 보여 줍니다.

AWS CLI

연결에 대한 정보를 가져오려면

다음 get-connection 예제에서는 연결에 대한 세부 정보를 보여줍니다.

```
aws codestar-connections get-connection \
    --connection-arn arn:aws:codestar-connections:us-east-1:123456789012:connection/
aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f
```

출력:

```
{
    "Connection": {
        "ConnectionName": "MyConnection",
        "ConnectionArn": "arn:aws:codestar-connections:us-
east-1:123456789012:connection/aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f",
        "ProviderType": "Bitbucket",
        "OwnerAccountId": "123456789012",
        "ConnectionStatus": "AVAILABLE"
    }
}
```

자세한 내용은 개발자 도구 콘솔 사용 설명서의 연결 세부 정보 보기를 참조하세요.

- 자세한 API 내용은 명령 참조GetConnection의 섹션을 참조하세요. AWS CLI

### get-host

다음 코드 예시에서는 get-host을 사용하는 방법을 보여 줍니다.

AWS CLI

호스트에 대한 정보를 가져오려면

다음 `get-host` 예제에서는 호스트에 대한 세부 정보를 보여줍니다.

```
aws codestar-connections get-host \
    --host-arn arn:aws:codestar-connections:us-east-1:123456789012:host/
MyHost-28aef605
```

출력:

```
{
    "Name": "MyHost",
    "Status": "AVAILABLE",
    "ProviderType": "GitHubEnterpriseServer",
    "ProviderEndpoint": "https://test-instance-1.dev/"
}
```

자세한 내용은 개발자 도구 콘솔 사용 설명서의 호스트 세부 정보 보기(CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조GetHost의 섹션을 참조하세요. AWS CLI

## `list-connections`

다음 코드 예시에서는 `list-connections`을 사용하는 방법을 보여 줍니다.

AWS CLI

연결을 나열하려면

다음 `list-connections` 예제에서는 Bitbucket 공급자 유형에 대한 계정의 모든 연결 목록을 검색합니다.

```
aws codestar-connections list-connections \
--provider-type Bitbucket \
--max-results 5 \
--next-token: next-token
```

출력:

```
{
```

```
    "Connections": [
        {
            "ConnectionName": "my-connection",
            "ProviderType": "Bitbucket",
            "Status": "PENDING",
            "ARN": "arn:aws:codestar-connections:us-east-1:123456789012:connection/
aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f",
            "OwnerAccountId": "123456789012"
        },
        {
            "ConnectionName": "my-other-connection",
            "ProviderType": "Bitbucket",
            "Status": "AVAILABLE",
            "ARN": "arn:aws:codestar-connections:us-east-1:123456789012:connection/
aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f",
            "OwnerAccountId": "123456789012"
        },
    ],
    "NextToken": "next-token"
}
```

자세한 내용은 개발자 도구 콘솔 사용 설명서의 연결 목록(CLI)을 참조하세요.

- 자세한 API 내용은 명령 참조ListConnections의 섹션을 참조하세요. AWS CLI

### `list-hosts`

다음 코드 예시에서는 `list-hosts`을 사용하는 방법을 보여 줍니다.

AWS CLI

호스트를 나열하려면

다음 `list-hosts` 예제에서는 계정의 모든 호스트 목록을 검색합니다.

```
aws codestar-connections list-hosts
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 개발자 도구 콘솔 사용 설명서의 호스트 목록(CLI)을 참조하세요.

- 자세한 API 내용은 명령 참조ListHosts의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 지정된 연결 리소스에 연결된 모든 태그 목록을 검색합니다.

```
aws codestar-connections list-tags-for-resource \
    --resource-arn arn:aws:codestar-connections:us-east-1:123456789012:connection/
aEXAMPLE-8aad-4d5d-8878-dfcab0bc441f
```

출력:

```
{
    "Tags": [
        {
            "Key": "Project",
            "Value": "ProjectA"
        },
        {
            "Key": "ReadOnly",
            "Value": "true"
        }
    ]
}
```

자세한 내용은 개발자 도구 콘솔 사용 설명서의 연결 리소스에 대한 태그 보기를 참조하세요.

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### tag-resource

다음 코드 예시에서는 tag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 태그를 지정하려면

다음 tag-resource 예제에서는 제공된 태그 세트를 연결과 연결합니다. 태그를 추가하거나 편집하려면 이 명령을 사용합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 개발자 도구 콘솔 사용 설명서의 연결 리소스에 태그 추가를 참조하세요.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### untag-resource

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

연결 리소스에서 AWS 태그를 제거하려면

다음은 지정된 리소스에서 태그를 untag-resource 제거합니다.

```
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
```

출력:

```
{
    "Tags": []
}
```

자세한 내용은 개발자 도구 콘솔 사용 설명서의 연결 리소스에서 태그 제거를 참조하세요.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

## 를 사용한 Amazon Cognito 자격 증명 예제 AWS CLI

다음 코드 예제에서는 Amazon Cognito Identity와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `create-identity-pool`

다음 코드 예시에서는 `create-identity-pool`을 사용하는 방법을 보여 줍니다.

AWS CLI

Cognito 자격 증명 풀 공급자를 사용하여 자격 증명 풀 생성

이 예제에서는 라는 이름의 자격 증명 풀을 생성합니다 MyIdentityPool. Cognito 자격 증명 풀 공급자가 있습니다. 인증되지 않은 자격 증명은 허용되지 않습니다.

명령:

```
aws cognito-identity create-identity-pool --identity-pool-
name MyIdentityPool --no-allow-unauthenticated-identities --cognito-
```

```
identity-providers ProviderName="cognito-idp.us-west-2.amazonaws.com/us-
west-2_aaaaaaaaa",ClientId="3n4b5urk1ft4fl3mg5e62d9ado",ServerSideTokenCheck=false
```

출력:

```
{
  "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
  "IdentityPoolName": "MyIdentityPool",
  "AllowUnauthenticatedIdentities": false,
  "CognitoIdentityProviders": [
      {
          "ProviderName": "cognito-idp.us-west-2.amazonaws.com/us-west-2_111111111",
          "ClientId": "3n4b5urk1ft4fl3mg5e62d9ado",
          "ServerSideTokenCheck": false
      }
  ]
}
```

- 자세한 API 내용은 명령 참조CreateIdentityPool의 섹션을 참조하세요. AWS CLI

### `delete-identities`

다음 코드 예시에서는 `delete-identities`을 사용하는 방법을 보여 줍니다.

AWS CLI

자격 증명 풀 삭제

이 예제에서는 자격 증명 풀을 삭제합니다.

명령:

```
aws cognito-identity delete-identity-pool --identity-ids-to-delete "us-
west-2:11111111-1111-1111-1111-111111111111"
```

출력:

```
{
  "UnprocessedIdentityIds": []
}
```

- 자세한 API 내용은 명령 참조DeleteIdentities의 섹션을 참조하세요. AWS CLI

### delete-identity-pool

다음 코드 예시에서는 delete-identity-pool을 사용하는 방법을 보여 줍니다.

AWS CLI

자격 증명 풀 삭제

다음 delete-identity-pool 예시에서는 지정된 자격 증명 풀을 삭제합니다.

명령:

```
aws cognito-identity delete-identity-pool \
    --identity-pool-id "us-west-2:11111111-1111-1111-1111-111111111111"
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeleteIdentityPool의 섹션을 참조하세요. AWS CLI

### describe-identity-pool

다음 코드 예시에서는 describe-identity-pool을 사용하는 방법을 보여 줍니다.

AWS CLI

자격 증명 풀을 설명하려면

이 예제에서는 자격 증명 풀을 설명합니다.

명령:

```
aws cognito-identity describe-identity-pool --identity-pool-id "us-
west-2:11111111-1111-1111-1111-111111111111"
```

출력:

```
{
  "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
  "IdentityPoolName": "MyIdentityPool",
  "AllowUnauthenticatedIdentities": false,
  "CognitoIdentityProviders": [
      {
```

```
            "ProviderName": "cognito-idp.us-west-2.amazonaws.com/us-west-2_111111111",
            "ClientId": "3n4b5urk1ft4fl3mg5e62d9ado",
            "ServerSideTokenCheck": false
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeIdentityPool의 섹션을 참조하세요. AWS CLI

### `get-identity-pool-roles`

다음 코드 예시에서는 `get-identity-pool-roles`을 사용하는 방법을 보여 줍니다.

AWS CLI

자격 증명 풀 역할을 가져오려면

이 예제에서는 자격 증명 풀 역할을 가져옵니다.

명령:

```
aws cognito-identity get-identity-pool-roles --identity-pool-id "us-
west-2:11111111-1111-1111-1111-111111111111"
```

출력:

```
{
  "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"
  }
}
```

- 자세한 API 내용은 명령 참조GetIdentityPoolRoles의 섹션을 참조하세요. AWS CLI

### `list-identity-pools`

다음 코드 예시에서는 `list-identity-pools`을 사용하는 방법을 보여 줍니다.

AWS CLI

자격 증명 풀 나열

이 예시에는 자격 증명 풀이 나열되어 있습니다. 최대 20개의 자격 증명이 나열되어 있습니다.

명령:

```
aws cognito-identity list-identity-pools --max-results 20
```

출력:

```
{
  "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"
      }
  ]
}
```

- 자세한 API 내용은 명령 참조ListIdentityPools의 섹션을 참조하세요. AWS CLI

### set-identity-pool-roles

다음 코드 예시에서는 set-identity-pool-roles을 사용하는 방법을 보여 줍니다.

AWS CLI

자격 증명 풀 역할을 설정하려면

다음 set-identity-pool-roles 예제에서는 자격 증명 풀 역할을 설정합니다.

```
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"
```

- 자세한 API 내용은 명령 참조SetIdentityPoolRoles의 섹션을 참조하세요. AWS CLI

### update-identity-pool

다음 코드 예시에서는 update-identity-pool을 사용하는 방법을 보여 줍니다.

AWS CLI

자격 증명 풀을 업데이트하려면

이 예제에서는 자격 증명 풀을 업데이트합니다. 이름을 로 설정합니다 MyIdentityPool. Cognito를 자격 증명 공급자로 추가합니다. 인증되지 않은 자격 증명은 허용되지 않습니다.

명령:

```
aws cognito-identity update-identity-pool --identity-pool-id "us-
west-2:11111111-1111-1111-1111-111111111111" --identity-pool-
name "MyIdentityPool" --no-allow-unauthenticated-identities --cognito-
identity-providers ProviderName="cognito-idp.us-west-2.amazonaws.com/us-
west-2_111111111",ClientId="3n4b5urk1ft4fl3mg5e62d9ado",ServerSideTokenCheck=false
```

출력:

```
{
  "IdentityPoolId": "us-west-2:11111111-1111-1111-1111-111111111111",
  "IdentityPoolName": "MyIdentityPool",
  "AllowUnauthenticatedIdentities": false,
  "CognitoIdentityProviders": [
      {
          "ProviderName": "cognito-idp.us-west-2.amazonaws.com/us-west-2_111111111",
          "ClientId": "3n4b5urk1ft4fl3mg5e62d9ado",
          "ServerSideTokenCheck": false
      }
  ]
}
```

- 자세한 API 내용은 명령 참조UpdateIdentityPool의 섹션을 참조하세요. AWS CLI

# 를 사용한 Amazon Cognito Identity Provider 예제 AWS CLI

다음 코드 예제에서는 Amazon Cognito Identity Provider와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `add-custom-attributes`

다음 코드 예시에서는 `add-custom-attributes`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 속성을 추가하려면

이 예제에서는 사용자 속성 CustomAttr1을 사용자 풀에 추가합니다. 문자열 유형이며 최소 1자에서 최대 15자여야 합니다. 이 값은 필수가 아닙니다.

명령:

```
aws cognito-idp add-custom-attributes --user-pool-id us-west-2_aaaaaaaaa --custom-
attributes
 Name="CustomAttr1",AttributeDataType="String",DeveloperOnlyAttribute=false,Required=false,S
```

- 자세한 API 내용은 명령 참조AddCustomAttributes의 섹션을 참조하세요. AWS CLI

### `admim-disable-user`

다음 코드 예시에서는 `admim-disable-user`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자를 비활성화하려면

이 예제에서는 사용자 jane@example.com을 비활성화합니다.

명령:

```
aws cognito-idp admin-disable-user --user-pool-id us-west-2_aaaaaaaaa --
username jane@example.com
```

- 자세한 API 내용은 명령 참조AdmimDisableUser의 섹션을 참조하세요. AWS CLI

### admim-enable-user

다음 코드 예시에서는 admim-enable-user을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자를 활성화하려면

이 예제에서는 사용자 이름 jane@example.com을 활성화합니다.

명령:

```
aws cognito-idp admin-enable-user --user-pool-id us-west-2_aaaaaaaaa --
username jane@example.com
```

- 자세한 API 내용은 명령 참조AdmimEnableUser의 섹션을 참조하세요. AWS CLI

### admin-add-user-to-group

다음 코드 예시에서는 admin-add-user-to-group을 사용하는 방법을 보여 줍니다.

AWS CLI

그룹에 사용자를 추가하려면

이 예제에서는 그룹 에 사용자 Jane을 추가합니다 MyGroup.

명령:

```
aws cognito-idp admin-add-user-to-group --user-pool-id us-west-2_aaaaaaaaa --
username Jane --group-name MyGroup
```

- 자세한 API 내용은 명령 참조AdminAddUserToGroup의 섹션을 참조하세요. AWS CLI

### admin-confirm-sign-up

다음 코드 예시에서는 admin-confirm-sign-up을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 등록을 확인하려면

이 예제에서는 사용자 jane@example.com을 확인합니다.

명령:

```
aws cognito-idp admin-confirm-sign-up --user-pool-id us-west-2_aaaaaaaaa --
username jane@example.com
```

- 자세한 API 내용은 명령 참조AdminConfirmSignUp의 섹션을 참조하세요. AWS CLI

### admin-create-user

다음 코드 예시에서는 admin-create-user을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자를 생성하려면

다음 admin-create-user 예제에서는 지정된 설정 이메일 주소와 전화번호를 사용하여 사용자를 생성합니다.

```
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
```

출력:

```
{
    "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"
    }
}
```

- 자세한 API 내용은 명령 참조AdminCreateUser의 섹션을 참조하세요. AWS CLI

### admin-delete-user-attributes

다음 코드 예시에서는 admin-delete-user-attributes을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 속성을 삭제하려면

이 예제에서는 사용자 diego@example.com에 대한 사용자 지정 속성 CustomAttr1을 삭제합니다.

명령:

```
aws cognito-idp admin-delete-user-attributes --user-pool-id us-west-2_aaaaaaaaa --
username diego@example.com --user-attribute-names "custom:CustomAttr1"
```

- 자세한 API 내용은 명령 참조AdminDeleteUserAttributes의 섹션을 참조하세요. AWS CLI

### admin-delete-user

다음 코드 예시에서는 admin-delete-user을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 삭제

이 예시는 사용자를 삭제합니다.

명령:

```
aws cognito-idp admin-delete-user --user-pool-id us-west-2_aaaaaaaaa --
username diego@example.com
```

- 자세한 API 내용은 명령 참조AdminDeleteUser의 섹션을 참조하세요. AWS CLI

### admin-forget-device

다음 코드 예시에서는 admin-forget-device을 사용하는 방법을 보여 줍니다.

AWS CLI

디바이스를 잊으려면

이 예제에서는 사용자 이름 jane@example.com의 디바이스를 잊어버렸습니다.

명령:

```
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
```

- 자세한 API 내용은 명령 참조AdminForgetDevice의 섹션을 참조하세요. AWS CLI

### admin-get-device

다음 코드 예시에서는 admin-get-device을 사용하는 방법을 보여 줍니다.

AWS CLI

디바이스를 가져오려면

이 예제에서는 사용자 이름 jane@example.com용 디바이스를 가져옵니다.

명령:

```
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
```

- 자세한 API 내용은 명령 참조AdminGetDevice의 섹션을 참조하세요. AWS CLI

### admin-get-user

다음 코드 예시에서는 admin-get-user을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 가져오기

이 예시에서는 사용자 이름 jane@example.com에 대한 정보를 가져옵니다.

명령:

```
aws cognito-idp admin-get-user --user-pool-id us-west-2_aaaaaaaaa --
username jane@example.com
```

출력:

```
{
  "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
}
```

- 자세한 API 내용은 명령 참조AdminGetUser의 섹션을 참조하세요. AWS CLI

### admin-initiate-auth

다음 코드 예시에서는 admin-initiate-auth을 사용하는 방법을 보여 줍니다.

AWS CLI

권한 부여 시작

이 예제에서는 사용자 이름 jane@example.com에 대한 ADMIN_NO_SRP_AUTH 흐름을 사용하여 권한 부여를 시작합니다.

클라이언트에는 서버 기반 인증(ADMIN_NO_SRP_AUTH)에 API 대한 로그인이 활성화되어 있어야 합니다.

반환 값의 세션 정보를 사용하여 admin-respond-to-auth-challenge를 호출합니다.

명령:

```
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
```

출력:

```
{
  "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\"}"
  }
}
```

- 자세한 API 내용은 명령 참조AdminInitiateAuth의 섹션을 참조하세요. AWS CLI

### admin-list-devices

다음 코드 예시에서는 admin-list-devices을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자의 디바이스를 나열하려면

이 예제에서는 사용자 이름 jane@example.com의 디바이스를 나열합니다.

명령:

```
aws cognito-idp admin-list-devices --user-pool-id us-west-2_aaaaaaaaa --
username jane@example.com
```

- 자세한 API 내용은 명령 참조AdminListDevices의 섹션을 참조하세요. AWS CLI

### admin-list-groups-for-user

다음 코드 예시에서는 admin-list-groups-for-user을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자의 그룹을 나열하려면

이 예제에서는 사용자 이름 jane@example.com의 그룹을 나열합니다.

명령:

```
aws cognito-idp admin-list-groups-for-user --user-pool-id us-west-2_aaaaaaaaa --
username diego@example.com
```

출력:

```
{
  "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
      }
  ]
}
```

- 자세한 API 내용은 명령 참조AdminListGroupsForUser의 섹션을 참조하세요. AWS CLI

### admin-list-user-auth-events

다음 코드 예시에서는 admin-list-user-auth-events을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자에 대한 권한 부여 이벤트를 나열하려면

이 예제에서는 사용자 이름 diego@example.com에 대한 권한 부여 이벤트를 나열합니다.

명령:

```
aws cognito-idp admin-list-user-auth-events --user-pool-id us-west-2_aaaaaaaaa --
username diego@example.com
```

- 자세한 API 내용은 명령 참조AdminListUserAuthEvents의 섹션을 참조하세요. AWS CLI

### admin-remove-user-from-group

다음 코드 예시에서는 admin-remove-user-from-group을 사용하는 방법을 보여 줍니다.

AWS CLI

그룹에서 사용자를 제거하려면

이 예제에서는 에서 jane@example.com을 제거합니다 SampleGroup.

명령:

```
aws cognito-idp admin-remove-user-from-group --user-pool-id us-west-2_aaaaaaaaa --
username jane@example.com --group-name SampleGroup
```

- 자세한 API 내용은 명령 참조AdminRemoveUserFromGroup의 섹션을 참조하세요. AWS CLI

### admin-reset-user-password

다음 코드 예시에서는 admin-reset-user-password을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 암호를 재설정하려면

이 예제에서는 diego@example.com의 암호를 재설정합니다.

명령:

```
aws cognito-idp admin-reset-user-password --user-pool-id us-west-2_aaaaaaaaa --
username diego@example.com
```

- 자세한 API 내용은 명령 참조AdminResetUserPassword의 섹션을 참조하세요. AWS CLI

### admin-set-user-mfa-preference

다음 코드 예시에서는 admin-set-user-mfa-preference을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 MFA 기본 설정을 지정하려면

이 예제에서는 사용자 이름 diego@example.com에 대한 SMS MFA 기본 설정을 지정합니다.

명령:

```
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
```

- 자세한 API 내용은 명령 참조AdminSetUserMfaPreference의 섹션을 참조하세요. AWS CLI

### admin-set-user-settings

다음 코드 예시에서는 admin-set-user-settings을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 설정을 설정하려면

이 예제에서는 사용자 이름 diego@example.com의 MFA 전송 기본 설정을 로 설정합니다EMAIL.

명령:

```
aws cognito-idp admin-set-user-settings --user-pool-id us-west-2_aaaaaaaaa --
username diego@example.com --mfa-options DeliveryMedium=EMAIL
```

- 자세한 API 내용은 명령 참조AdminSetUserSettings의 섹션을 참조하세요. AWS CLI

### admin-update-auth-event-feedback

다음 코드 예시에서는 admin-update-auth-event-feedback을 사용하는 방법을 보여 줍니다.

AWS CLI

권한 부여 이벤트에 대한 피드백을 제공하려면

이 예제에서는 event-id로 식별된 권한 부여 이벤트에 대한 피드백 값을 Valid로 설정합니다.

명령:

```
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
```

- 자세한 API 내용은 명령 참조AdminUpdateAuthEventFeedback의 섹션을 참조하세요. AWS CLI

### admin-update-device-status

다음 코드 예시에서는 admin-update-device-status을 사용하는 방법을 보여 줍니다.

AWS CLI

디바이스 상태를 업데이트하려면

이 예제에서는 device-key로 식별된 디바이스의 디바이스 기억 상태를 not_remembered로 설정합니다.

명령:

```
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
```

- 자세한 API 내용은 명령 참조AdminUpdateDeviceStatus의 섹션을 참조하세요. AWS CLI

### admin-update-user-attributes

다음 코드 예시에서는 admin-update-user-attributes을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 속성을 업데이트하려면

이 예제에서는 사용자 diego@example.com에 대한 사용자 지정 사용자 속성 CustomAttr1을 업데이트합니다.

명령:

```
aws cognito-idp admin-update-user-attributes --user-pool-id us-
west-2_aaaaaaaaa --username diego@example.com --user-attributes
 Name="custom:CustomAttr1",Value="Purple"
```

- 자세한 API 내용은 명령 참조AdminUpdateUserAttributes의 섹션을 참조하세요. AWS CLI

### change-password

다음 코드 예시에서는 change-password을 사용하는 방법을 보여 줍니다.

AWS CLI

암호를 변경하려면

이 예제에서는 암호를 변경합니다.

명령:

```
aws cognito-idp change-password --previous-password OldPassword --proposed-
password NewPassword --access-token ACCESS_TOKEN
```

- 자세한 API 내용은 명령 참조ChangePassword의 섹션을 참조하세요. AWS CLI

### `confirm-forgot-password`

다음 코드 예시에서는 `confirm-forgot-password`을 사용하는 방법을 보여 줍니다.

AWS CLI

잊어버린 암호를 확인하려면

이 예제에서는 사용자 이름 diego@example.com의 암호를 잊어버렸음을 확인합니다.

명령:

```
aws cognito-idp confirm-forgot-password --client-id 3n4b5urk1ft4fl3mg5e62d9ado --
username=diego@example.com --password PASSWORD --confirmation-code CONF_CODE
```

- 자세한 API 내용은 명령 참조ConfirmForgotPassword의 섹션을 참조하세요. AWS CLI

### `confirm-sign-up`

다음 코드 예시에서는 `confirm-sign-up`을 사용하는 방법을 보여 줍니다.

AWS CLI

가입 확인

이 예시는 사용자 이름 diego@example.com 가입을 확인합니다.

명령:

```
aws cognito-idp confirm-sign-up --client-id 3n4b5urk1ft4fl3mg5e62d9ado --
username=diego@example.com --confirmation-code CONF_CODE
```

- 자세한 API 내용은 명령 참조ConfirmSignUp의 섹션을 참조하세요. AWS CLI

### create-group

다음 코드 예시에서는 create-group을 사용하는 방법을 보여 줍니다.

AWS CLI

그룹을 생성하려면

이 예제에서는 설명이 포함된 그룹을 생성합니다.

명령:

```
aws cognito-idp create-group --user-pool-id us-west-2_aaaaaaaaa --group-
name MyNewGroup --description "New group."
```

출력:

```
{
  "Group": {
      "GroupName": "MyNewGroup",
      "UserPoolId": "us-west-2_aaaaaaaaa",
      "Description": "New group.",
      "LastModifiedDate": 1548270073.795,
      "CreationDate": 1548270073.795
  }
}
```

역할 및 우선 순위가 있는 그룹을 생성하려면

이 예제에서는 설명이 포함된 그룹을 생성합니다. 여기에는 역할 및 우선순위도 포함됩니다.

명령:

```
aws cognito-idp create-group --user-pool-id us-west-2_aaaaaaaaa --group-
name MyNewGroupWithRole --description "New group with a role." --role-
arn arn:aws:iam::111111111111:role/MyNewGroupRole --precedence 2
```

출력:

```
{
  "Group": {
      "GroupName": "MyNewGroupWithRole",
```

```
      "UserPoolId": "us-west-2_aaaaaaaaa",
      "Description": "New group with a role.",
      "RoleArn": "arn:aws:iam::111111111111:role/MyNewGroupRole",
      "Precedence": 2,
      "LastModifiedDate": 1548270211.761,
      "CreationDate": 1548270211.761
  }
}
```

- 자세한 API 내용은 명령 참조CreateGroup의 섹션을 참조하세요. AWS CLI

### `create-user-import-job`

다음 코드 예시에서는 `create-user-import-job`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 가져오기 작업을 생성하려면

이 예제에서는 라는 사용자 가져오기 작업을 생성합니다 MyImportJob.

사용자 가져오기에 대한 자세한 내용은 CSV 파일에서 사용자 풀로 사용자 가져오기를 참조하세요.

명령:

```
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
```

출력:

```
{
  "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
    }
}
```

미리 서명된 를 사용하여 curl이 있는 .csv 파일을 업로드합니다URL.

명령:

```
curl -v -T "PATH_TO_CSV_FILE" -H "x-amz-server-side-encryption:aws:kms"
 "PRE_SIGNED_URL"
```

- 자세한 API 내용은 명령 참조CreateUserImportJob의 섹션을 참조하세요. AWS CLI

### `create-user-pool-client`

다음 코드 예시에서는 `create-user-pool-client`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 풀 클라이언트를 생성하려면

이 예제에서는 USER__ 및 _ADMINNOPASSWORD__AUTH라는 두 가지 명시적 권한 부여 흐름이 있는 새 사용자 풀 클라이언트를 생성합니다SRPAUTH.

명령:

```
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"
```

출력:

```
{
  "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
    }
}
```

- 자세한 API 내용은 명령 참조CreateUserPoolClient의 섹션을 참조하세요. AWS CLI

### `create-user-pool-domain`

다음 코드 예시에서는 `create-user-pool-domain`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 풀 도메인을 생성하려면

이 예제에서는 USER__ 및 _ADMINNOPASSWORD__AUTH라는 두 가지 명시적 권한 부여 흐름으로 새 사용자 풀 도메인을 생성합니다SRPAUTH.

명령:

```
aws cognito-idp create-user-pool-domain --user-pool-id us-west-2_aaaaaaaaa  --domain my-new-domain
```

- 자세한 API 내용은 명령 참조CreateUserPoolDomain의 섹션을 참조하세요. AWS CLI

### `create-user-pool`

다음 코드 예시에서는 `create-user-pool`을 사용하는 방법을 보여 줍니다.

AWS CLI

최소 구성 사용자 풀 생성

이 예제에서는 기본값을 MyUserPool 사용하여 라는 사용자 풀을 생성합니다. 필수 속성도 없고 애플리케이션 클라이언트도 없습니다. MFA 및 고급 보안이 비활성화되었습니다.

명령:

```
aws cognito-idp create-user-pool --pool-name MyUserPool
```

출력:

```
{
  "UserPool": {
      "SchemaAttributes": [
          {
              "Name": "sub",
              "StringAttributeConstraints": {
                  "MinLength": "1",
                  "MaxLength": "2048"
              },
              "DeveloperOnlyAttribute": false,
              "Required": true,
              "AttributeDataType": "String",
              "Mutable": false
          },
          {
              "Name": "name",
              "StringAttributeConstraints": {
                  "MinLength": "0",
                  "MaxLength": "2048"
              },
              "DeveloperOnlyAttribute": false,
              "Required": false,
              "AttributeDataType": "String",
              "Mutable": true
          },
          {
              "Name": "given_name",
              "StringAttributeConstraints": {
                  "MinLength": "0",
                  "MaxLength": "2048"
              },
              "DeveloperOnlyAttribute": false,
              "Required": false,
              "AttributeDataType": "String",
              "Mutable": true
          },
          {
              "Name": "family_name",
              "StringAttributeConstraints": {
                  "MinLength": "0",
                  "MaxLength": "2048"
              },
```

```
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "middle_name",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "nickname",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "preferred_username",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "profile",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
```

```
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "picture",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "website",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "email",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "AttributeDataType": "Boolean",
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "Name": "email_verified",
            "Mutable": true
```

```
        },
        {
            "Name": "gender",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "birthdate",
            "StringAttributeConstraints": {
                "MinLength": "10",
                "MaxLength": "10"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "zoneinfo",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "locale",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
```

```
        },
        {
            "Name": "phone_number",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "AttributeDataType": "Boolean",
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "Name": "phone_number_verified",
            "Mutable": true
        },
        {
            "Name": "address",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "updated_at",
            "NumberAttributeConstraints": {
                "MinValue": "0"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "Number",
            "Mutable": true
        }
    ],
    "MfaConfiguration": "OFF",
    "Name": "MyUserPool",
    "LastModifiedDate": 1547833345.777,
```

```
        "AdminCreateUserConfig": {
            "UnusedAccountValidityDays": 7,
            "AllowAdminCreateUserOnly": false
        },
        "EmailConfiguration": {},
        "Policies": {
            "PasswordPolicy": {
                "RequireLowercase": true,
                "RequireSymbols": true,
                "RequireNumbers": true,
                "MinimumLength": 8,
                "RequireUppercase": true
            }
        },
        "CreationDate": 1547833345.777,
        "EstimatedNumberOfUsers": 0,
        "Id": "us-west-2_aaaaaaaaa",
        "LambdaConfig": {}
    }
}
```

두 개의 필수 속성으로 사용자 풀을 생성하는 방법

이 예제에서는 사용자 풀을 생성합니다 MyUserPool. 풀은 이메일을 사용자 이름 속성으로 받아들이도록 구성되어 있습니다. 또한 Amazon Simple Email Service를 사용하여 이메일 소스 주소를 검증된 주소로 설정합니다.

명령:

```
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"
```

출력:

```
{
    "UserPool": {
        "SchemaAttributes": [
            {
                "Name": "sub",
                "StringAttributeConstraints": {
                    "MinLength": "1",
```

```
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": true,
            "AttributeDataType": "String",
            "Mutable": false
        },
        {
            "Name": "name",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "given_name",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "family_name",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "middle_name",
            "StringAttributeConstraints": {
                "MinLength": "0",
```

```
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "nickname",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "preferred_username",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "profile",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "picture",
            "StringAttributeConstraints": {
                "MinLength": "0",
```

```
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "website",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "email",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "AttributeDataType": "Boolean",
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "Name": "email_verified",
            "Mutable": true
        },
        {
            "Name": "gender",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
```

```
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "birthdate",
            "StringAttributeConstraints": {
                "MinLength": "10",
                "MaxLength": "10"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "zoneinfo",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "locale",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "phone_number",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
```

```
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "AttributeDataType": "Boolean",
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "Name": "phone_number_verified",
            "Mutable": true
        },
        {
            "Name": "address",
            "StringAttributeConstraints": {
                "MinLength": "0",
                "MaxLength": "2048"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "String",
            "Mutable": true
        },
        {
            "Name": "updated_at",
            "NumberAttributeConstraints": {
                "MinValue": "0"
            },
            "DeveloperOnlyAttribute": false,
            "Required": false,
            "AttributeDataType": "Number",
            "Mutable": true
        }
    ],
    "MfaConfiguration": "OFF",
    "Name": "MyUserPool",
    "LastModifiedDate": 1547837788.189,
    "AdminCreateUserConfig": {
        "UnusedAccountValidityDays": 7,
        "AllowAdminCreateUserOnly": false
    },
    "EmailConfiguration": {
        "ReplyToEmailAddress": "jane@example.com",
        "SourceArn": "arn:aws:ses:us-east-1:111111111111:identity/
jane@example.com"
    },
```

```
        "Policies": {
            "PasswordPolicy": {
                "RequireLowercase": true,
                "RequireSymbols": true,
                "RequireNumbers": true,
                "MinimumLength": 8,
                "RequireUppercase": true
            }
        },
        "UsernameAttributes": [
            "email"
        ],
        "CreationDate": 1547837788.189,
        "EstimatedNumberOfUsers": 0,
        "Id": "us-west-2_aaaaaaaaa",
        "LambdaConfig": {}
    }
}
```

- 자세한 API 내용은 명령 참조CreateUserPool의 섹션을 참조하세요. AWS CLI

### delete-group

다음 코드 예시에서는 delete-group을 사용하는 방법을 보여 줍니다.

AWS CLI

그룹을 삭제하려면

이 예제에서는 그룹을 삭제합니다.

명령:

```
aws cognito-idp delete-group --user-pool-id us-west-2_aaaaaaaaa  --group-
name MyGroupName
```

- 자세한 API 내용은 명령 참조DeleteGroup의 섹션을 참조하세요. AWS CLI

### delete-identity-provider

다음 코드 예시에서는 delete-identity-provider을 사용하는 방법을 보여 줍니다.

AWS CLI

자격 증명 공급자를 삭제하려면

이 예제에서는 자격 증명 공급자를 삭제합니다.

명령:

```
aws cognito-idp delete-identity-provider --user-pool-id us-west-2_aaaaaaaaa --
provider-name Facebook
```

- 자세한 API 내용은 명령 참조DeleteIdentityProvider의 섹션을 참조하세요. AWS CLI

### `delete-resource-server`

다음 코드 예시에서는 `delete-resource-server`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 서버를 삭제하려면

이 예제에서는 weather.example.com 리소스 서버를 삭제합니다.

명령:

```
aws cognito-idp delete-resource-server --user-pool-id us-west-2_aaaaaaaaa --
identifier weather.example.com
```

- 자세한 API 내용은 명령 참조DeleteResourceServer의 섹션을 참조하세요. AWS CLI

### `delete-user-attributes`

다음 코드 예시에서는 `delete-user-attributes`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 속성을 삭제하려면

이 예제에서는 사용자 속성 'FAVORITE_ANIMAL'를 삭제합니다.

명령:

```
aws cognito-idp delete-user-attributes --access-token ACCESS_TOKEN --user-attribute-names "FAVORITE_ANIMAL"
```

- 자세한 API 내용은 명령 참조DeleteUserAttributes의 섹션을 참조하세요. AWS CLI

### delete-user-pool-client

다음 코드 예시에서는 delete-user-pool-client을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 풀 클라이언트를 삭제하려면

이 예제에서는 사용자 풀 클라이언트를 삭제합니다.

명령:

```
aws cognito-idp delete-user-pool-client --user-pool-id us-west-2_aaaaaaaaa --client-id 38fjsnc484p94kpqsnet7mpld0
```

- 자세한 API 내용은 명령 참조DeleteUserPoolClient의 섹션을 참조하세요. AWS CLI

### delete-user-pool-domain

다음 코드 예시에서는 delete-user-pool-domain을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 풀 도메인을 삭제하려면

다음 delete-user-pool-domain 예제에서는 라는 사용자 풀 도메인을 삭제합니다. my-domain

```
aws cognito-idp delete-user-pool-domain \
    --user-pool-id us-west-2_aaaaaaaaa \
    --domain my-domain
```

- 자세한 API 내용은 명령 참조DeleteUserPoolDomain의 섹션을 참조하세요. AWS CLI

### delete-user-pool

다음 코드 예시에서는 delete-user-pool을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 풀을 삭제하려면

이 예제에서는 사용자 풀 ID us-west-2_aaaaaaaa를 사용하여 사용자 풀을 삭제합니다.

명령:

```
aws cognito-idp delete-user-pool --user-pool-id us-west-2_aaaaaaaaa
```

- 자세한 API 내용은 명령 참조DeleteUserPool의 섹션을 참조하세요. AWS CLI

### delete-user

다음 코드 예시에서는 delete-user을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 삭제

이 예시는 사용자를 삭제합니다.

명령:

```
aws cognito-idp delete-user --access-token ACCESS_TOKEN
```

- 자세한 API 내용은 명령 참조DeleteUser의 섹션을 참조하세요. AWS CLI

### describe-identity-provider

다음 코드 예시에서는 describe-identity-provider을 사용하는 방법을 보여 줍니다.

AWS CLI

자격 증명 공급자를 설명하려면

이 예제에서는 Facebook이라는 자격 증명 공급자를 설명합니다.

명령:

```
aws cognito-idp describe-identity-provider --user-pool-id us-west-2_aaaaaaaaa --
provider-name Facebook
```

출력:

```
{
  "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
  }
}
```

- 자세한 API 내용은 명령 참조DescribeIdentityProvider의 섹션을 참조하세요. AWS CLI

### describe-resource-server

다음 코드 예시에서는 describe-resource-server을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 서버를 설명하려면

이 예제에서는 리소스 서버 weather.example.com 설명합니다.

명령:

```
aws cognito-idp describe-resource-server --user-pool-id us-west-2_aaaaaaaaa --
identifier weather.example.com
```

출력:

```
{
  "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"
          }
      ]
  }
}
```

- 자세한 API 내용은 명령 참조DescribeResourceServer의 섹션을 참조하세요. AWS CLI

### describe-risk-configuration

다음 코드 예시에서는 describe-risk-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

위험 구성을 설명하려면

이 예제에서는 us-west-2_aaaaaaaaa 풀과 관련된 위험 구성을 설명합니다.

명령:

```
aws cognito-idp describe-risk-configuration --user-pool-id us-west-2_aaaaaaaaa
```

출력:

```
{
  "RiskConfiguration": {
      "UserPoolId": "us-west-2_aaaaaaaaa",
      "CompromisedCredentialsRiskConfiguration": {
          "EventFilter": [
              "SIGN_IN",
              "SIGN_UP",
              "PASSWORD_CHANGE"
          ],
          "Actions": {
              "EventAction": "BLOCK"
          }
      },
      "AccountTakeoverRiskConfiguration": {
          "NotifyConfiguration": {
              "From": "diego@example.com",
              "ReplyTo": "diego@example.com",
              "SourceArn": "arn:aws:ses:us-east-1:111111111111:identity/
diego@example.com",
              "BlockEmail": {
                  "Subject": "Blocked sign-in attempt",
                  "HtmlBody": "<!DOCTYPE html>\n<html>\n<head>\n\t<title>HTML
 email context</title>\n\t<meta charset=\"utf-8\">\n</head>\n<body>\n<pre>We
 blocked an unrecognized sign-in to your account with this information:\n<ul>
\n<li>Time: {login-time}</li>\n<li>Device: {device-name}</li>\n<li>Location: {city},
 {country}</li>\n</ul>\nIf this sign-in was not by you, you should change your
 password and notify us by clicking on <a href={one-click-link-invalid}>this link</
a>\nIf this sign-in was by you, you can follow <a href={one-click-link-valid}>this
 link</a> to let us know</pre>\n</body>\n</html>",
                  "TextBody": "We blocked an unrecognized sign-in to your account
 with this information:\nTime: {login-time}\nDevice: {device-name}\nLocation:
 {city}, {country}\nIf this sign-in was not by you, you should change your password
 and notify us by clicking on {one-click-link-invalid}\nIf this sign-in was by you,
 you can follow {one-click-link-valid} to let us know"
              },
              "NoActionEmail": {
                  "Subject": "New sign-in attempt",
                  "HtmlBody": "<!DOCTYPE html>\n<html>\n<head>\n\t<title>HTML
 email context</title>\n\t<meta charset=\"utf-8\">\n</head>\n<body>\n<pre>We
 observed an unrecognized sign-in to your account with this information:\n<ul>
\n<li>Time: {login-time}</li>\n<li>Device: {device-name}</li>\n<li>Location: {city},
 {country}</li>\n</ul>\nIf this sign-in was not by you, you should change your
```

```
 password and notify us by clicking on <a href={one-click-link-invalid}>this link</
a>\nIf this sign-in was by you, you can follow <a href={one-click-link-valid}>this
 link</a> to let us know</pre>\n</body>\n</html>",
                    "TextBody": "We observed an unrecognized sign-in to your account
 with this information:\nTime: {login-time}\nDevice: {device-name}\nLocation:
 {city}, {country}\nIf this sign-in was not by you, you should change your password
 and notify us by clicking on {one-click-link-invalid}\nIf this sign-in was by you,
 you can follow {one-click-link-valid} to let us know"
                },
                "MfaEmail": {
                    "Subject": "New sign-in attempt",
                    "HtmlBody": "<!DOCTYPE html>\n<html>\n<head>\n\t<title>HTML email
 context</title>\n\t<meta charset=\"utf-8\">\n</head>\n<body>\n<pre>We required
 you to use multi-factor authentication for the following sign-in attempt:\n<ul>
\n<li>Time: {login-time}</li>\n<li>Device: {device-name}</li>\n<li>Location: {city},
 {country}</li>\n</ul>\nIf this sign-in was not by you, you should change your
 password and notify us by clicking on <a href={one-click-link-invalid}>this link</
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 required you to use multi-factor authentication
 for the following sign-in attempt:\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"
                }
            },
            "Actions": {
                "LowAction": {
                    "Notify": true,
                    "EventAction": "NO_ACTION"
                },
                "MediumAction": {
                    "Notify": true,
                    "EventAction": "MFA_IF_CONFIGURED"
                },
                "HighAction": {
                    "Notify": true,
                    "EventAction": "MFA_IF_CONFIGURED"
                }
            }
        }
    }
}
```

- 자세한 API 내용은 명령 참조DescribeRiskConfiguration의 섹션을 참조하세요. AWS CLI

### describe-user-import-job

다음 코드 예시에서는 describe-user-import-job을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 가져오기 작업을 설명하려면

이 예제에서는 사용자 입력 작업에 대해 설명합니다.

사용자 가져오기에 대한 자세한 내용은 CSV 파일에서 사용자 풀로 사용자 가져오기를 참조하세요.

명령:

```
aws cognito-idp describe-user-import-job --user-pool-id us-west-2_aaaaaaaaa --job-
id import-TZqNQvDRnW
```

출력:

```
{
  "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
  }
}
```

- 자세한 API 내용은 명령 참조DescribeUserImportJob의 섹션을 참조하세요. AWS CLI

### describe-user-pool-client

다음 코드 예시에서는 describe-user-pool-client을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 풀 클라이언트를 설명하려면

이 예제에서는 사용자 풀 클라이언트를 설명합니다.

명령:

```
aws cognito-idp describe-user-pool-client --user-pool-id us-west-2_aaaaaaaaa --
client-id 38fjsnc484p94kpqsnet7mpld0
```

출력:

```
{
  "UserPoolClient": {
      "UserPoolId": "us-west-2_aaaaaaaaa",
      "ClientName": "MyApp",
      "ClientId": "38fjsnc484p94kpqsnet7mpld0",
      "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
    }
}
```

- 자세한 API 내용은 명령 참조DescribeUserPoolClient의 섹션을 참조하세요. AWS CLI

### `describe-user-pool-domain`

다음 코드 예시에서는 `describe-user-pool-domain`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 풀 클라이언트를 설명하려면

이 예제에서는 my-domain이라는 사용자 풀 도메인을 설명합니다.

명령:

```
aws cognito-idp describe-user-pool-domain --domain my-domain
```

출력:

```
{
  "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": {}
  }
}
```

- 자세한 API 내용은 명령 참조DescribeUserPoolDomain의 섹션을 참조하세요. AWS CLI

### describe-user-pool

다음 코드 예시에서는 describe-user-pool을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 풀을 설명하려면

이 예제에서는 사용자 풀 ID us-west-2_aaaaaaaaaa를 사용하는 사용자 풀을 설명합니다.

명령:

```
aws cognito-idp describe-user-pool --user-pool-id us-west-2_aaaaaaaaa
```

출력:

```
{
  "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": {}
    }
}
```

- 자세한 API 내용은 명령 참조DescribeUserPool의 섹션을 참조하세요. AWS CLI

### `forget-device`

다음 코드 예시에서는 `forget-device`을 사용하는 방법을 보여 줍니다.

AWS CLI

디바이스를 잊으려면

이 예제에서는 디바이스를 잊어버립니다.

명령:

```
aws cognito-idp forget-device --device-key us-west-2_abcd_1234-5678
```

- 자세한 API 내용은 명령 참조ForgetDevice의 섹션을 참조하세요. AWS CLI

### forgot-password

다음 코드 예시에서는 forgot-password을 사용하는 방법을 보여 줍니다.

AWS CLI

암호를 강제로 변경하려면

다음 forgot-password 예제에서는 암호를 변경하라는 메시지를 jane@example.com으로 보냅니다.

```
aws cognito-idp forgot-password --client-id 38fjsnc484p94kpqsnet7mpld0 --
username jane@example.com
```

출력:

```
{
    "CodeDeliveryDetails": {
        "Destination": "j***@e***.com",
        "DeliveryMedium": "EMAIL",
        "AttributeName": "email"
    }
}
```

- 자세한 API 내용은 명령 참조ForgotPassword의 섹션을 참조하세요. AWS CLI

### get-csv-header

다음 코드 예시에서는 get-csv-header을 사용하는 방법을 보여 줍니다.

AWS CLI

csv 헤더를 생성하려면

이 예제에서는 csv 헤더를 생성합니다.

사용자 가져오기에 대한 자세한 내용은 CSV 파일에서 사용자 풀로 사용자 가져오기를 참조하세요.

명령:

```
aws cognito-idp get-csv-header --user-pool-id us-west-2_aaaaaaaaa
```

출력:

```
{
  "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"
  ]
}
```

... CSV 파일에서 사용자 풀로 사용자 가져오기: https://docs.aws.amazon.com/cognito/latest/developerguide/cognito-user-pools-using-import-tool.html

- 자세한 API 내용은 명령 참조GetCsvHeader의 섹션을 참조하세요. AWS CLI

### get-group

다음 코드 예시에서는 get-group을 사용하는 방법을 보여 줍니다.

AWS CLI

그룹에 대한 정보를 가져오려면

이 예제에서는 라는 이름의 그룹에 대한 정보를 가져옵니다 MyGroup.

명령:

```
aws cognito-idp get-group --user-pool-id us-west-2_aaaaaaaaa --group-name MyGroup
```

출력:

```
{
  "Group": {
      "GroupName": "MyGroup",
      "UserPoolId": "us-west-2_aaaaaaaaa",
      "Description": "A sample group.",
      "LastModifiedDate": 1548270073.795,
      "CreationDate": 1548270073.795
  }
}
```

- 자세한 API 내용은 명령 참조GetGroup의 섹션을 참조하세요. AWS CLI

### get-signing-certificate

다음 코드 예시에서는 get-signing-certificate을 사용하는 방법을 보여 줍니다.

AWS CLI

서명 인증서를 가져오려면

이 예제에서는 사용자 풀에 대한 서명 인증서를 가져옵니다.

명령:

```
aws cognito-idp get-signing-certificate --user-pool-id us-west-2_aaaaaaaaa
```

출력:

```
{
  "Certificate": "CERTIFICATE_DATA"
}
```

- 자세한 API 내용은 명령 참조GetSigningCertificate의 섹션을 참조하세요. AWS CLI

### get-ui-customization

다음 코드 예시에서는 get-ui-customization을 사용하는 방법을 보여 줍니다.

AWS CLI

UI 사용자 지정 정보를 가져오려면

이 예제에서는 사용자 풀에 대한 UI 사용자 지정 정보를 가져옵니다.

명령:

```
aws cognito-idp get-ui-customization --user-pool-id us-west-2_aaaaaaaaa
```

출력:

```
{
  "UICustomization": {
      "UserPoolId": "us-west-2_aaaaaaaaa",
      "ClientId": "ALL",
      "ImageUrl": "https://aaaaaaaaaaaaa.cloudfront.net/us-west-2_aaaaaaaaa/
ALL/20190128231240/assets/images/image.jpg",
      "CSS": ".logo-customizable {\n\tmax-width: 60%;\n\tmax-height: 30%;
\n}\n.banner-customizable {\n\tpadding: 25px 0px 25px 10px;\n\tbackground-color:
 lightgray;\n}\n.label-customizable {\n\tfont-weight: 300;\n}\n.textDescription-
customizable {\n\tpadding-top: 10px;\n\tpadding-bottom: 10px;\n\tdisplay: block;
\n\tfont-size: 16px;\n}\n.idpDescription-customizable {\n\tpadding-top: 10px;\n
\tpadding-bottom: 10px;\n\tdisplay: block;\n\tfont-size: 16px;\n}\n.legalText-
customizable {\n\tcolor: #747474;\n\tfont-size: 11px;\n}\n.submitButton-customizable
 {\n\tfont-size: 14px;\n\tfont-weight: bold;\n\tmargin: 20px 0px 10px 0px;\n
\theight: 40px;\n\twidth: 100%;\n\tcolor: #fff;\n\tbackground-color: #337ab7;
\n}\n.submitButton-customizable:hover {\n\tcolor: #fff;\n\tbackground-color:
 #286090;\n}\n.errorMessage-customizable {\n\tpadding: 5px;\n\tfont-size: 14px;
```

```
\n\twidth: 100%;\n\tbackground: #F5F5F5;\n\tborder: 2px solid #D64958;\n\tcolor:
 #D64958;\n}\n.inputField-customizable {\n\twidth: 100%;\n\theight: 34px;\n\tcolor:
 #555;\n\tbackground-color: #fff;\n\tborder: 1px solid #ccc;\n}\n.inputField-
customizable:focus {\n\tborder-color: #66afe9;\n\toutline: 0;\n}\n.idpButton-
customizable {\n\theight: 40px;\n\twidth: 100%;\n\ttext-align: center;\n\tmargin-
bottom: 15px;\n\tcolor: #fff;\n\tbackground-color: #5bc0de;\n\tborder-color:
 #46b8da;\n}\n.idpButton-customizable:hover {\n\tcolor: #fff;\n\tbackground-color:
 #31b0d5;\n}\n.socialButton-customizable {\n\theight: 40px;\n\ttext-align: left;
\n\twidth: 100%;\n\tmargin-bottom: 15px;\n}\n.redirect-customizable {\n\ttext-
align: center;\n}\n.passwordCheck-notValid-customizable {\n\tcolor: #DF3312;
\n}\n.passwordCheck-valid-customizable {\n\tcolor: #19BF00;\n}\n.background-
customizable {\n\tbackground-color: #faf;\n}\n",
      "CSSVersion": "20190128231240"
  }
}
```

- 자세한 API 내용은 명령 참조GetUiCustomization의 섹션을 참조하세요. AWS CLI

### `list-user-import-jobs`

다음 코드 예시에서는 `list-user-import-jobs`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 가져오기 작업을 나열하려면

이 예제에서는 사용자 가져오기 작업을 나열합니다.

사용자 가져오기에 대한 자세한 내용은 CSV 파일에서 사용자 풀로 사용자 가져오기를 참조하세요.

명령:

```
aws cognito-idp list-user-import-jobs --user-pool-id us-west-2_aaaaaaaaa  --max-
results 20
```

출력:

```
{
  "UserImportJobs": [
      {
          "JobName": "Test2",
          "JobId": "import-d0OnwGA3mV",
```

```
            "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."
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListUserImportJobs의 섹션을 참조하세요. AWS CLI

### `list-user-pools`

다음 코드 예시에서는 `list-user-pools`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 풀 나열

이 예시에서는 최대 20개의 사용자 풀을 나열합니다.

명령:

```
aws cognito-idp list-user-pools --max-results 20
```

출력:

```
{
  "UserPools": [
      {
         "CreationDate": 1547763720.822,
         "LastModifiedDate": 1547763720.822,
         "LambdaConfig": {},
         "Id": "us-west-2_aaaaaaaaa",
         "Name": "MyUserPool"
      }
  ]
}
```

- 자세한 API 내용은 명령 참조ListUserPools의 섹션을 참조하세요. AWS CLI

### `list-users-in-group`

다음 코드 예시에서는 `list-users-in-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

그룹의 사용자를 나열하려면

이 예제에서는 그룹 의 사용자를 나열합니다 MyGroup.

명령:

```
aws cognito-idp list-users-in-group --user-pool-id us-west-2_aaaaaaaaa --group-
name MyGroup
```

출력:

```
{
  "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListUsersInGroup의 섹션을 참조하세요. AWS CLI

### `list-users`

다음 코드 예시에서는 `list-users`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 나열

이 예시에서는 최대 20개의 사용자를 나열합니다.

명령:

```
aws cognito-idp list-users --user-pool-id us-west-2_aaaaaaaaa --limit 20
```

출력:

```
{
  "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"
            }
        ]
    }
  ]
}
```

- 자세한 API 내용은 명령 참조ListUsers의 섹션을 참조하세요. AWS CLI

### `resend-confirmation-code`

다음 코드 예시에서는 `resend-confirmation-code`을 사용하는 방법을 보여 줍니다.

AWS CLI

확인 코드 다시 보내기

다음 `resend-confirmation-code` 예시에서는 사용자 `jane`에게 확인 코드를 보냅니다.

```
aws cognito-idp resend-confirmation-code \
    --client-id 12a3b456c7de890f11g123hijk \
    --username jane
```

출력:

```
{
    "CodeDeliveryDetails": {
        "Destination": "j***@e***.com",
        "DeliveryMedium": "EMAIL",
        "AttributeName": "email"
    }
}
```

자세한 내용은 Amazon Cognito 개발자 안내서의 사용자 계정 가입 및 확인 섹션을 참조하세요.

- 자세한 API 내용은 명령 참조ResendConfirmationCode의 섹션을 참조하세요. AWS CLI

### `respond-to-auth-challenge`

다음 코드 예시에서는 `respond-to-auth-challenge`을 사용하는 방법을 보여 줍니다.

AWS CLI

인증 문제에 응답

이 예시에서는 initiate-auth로 시작된 인증 문제에 응답합니다. 이는 NEW_PASSWORD_REQUIRED 챌린지에 대한 응답입니다. 사용자 jane@example.com의 암호를 설정합니다.

명령:

```
aws cognito-idp respond-to-auth-challenge --client-id 3n4b5urk1ft4fl3mg5e62d9ado
 --challenge-name NEW_PASSWORD_REQUIRED --challenge-responses
 USERNAME=jane@example.com,NEW_PASSWORD="password" --session "SESSION_TOKEN"
```

출력:

```
{
  "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"
      }
  }
}
```

- 자세한 API 내용은 명령 참조RespondToAuthChallenge의 섹션을 참조하세요. AWS CLI

### `set-risk-configuration`

다음 코드 예시에서는 `set-risk-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

위험 구성을 설정하려면

이 예제에서는 사용자 풀에 대한 위험 구성을 설정합니다. 가입 이벤트 작업을 NO_로 설정합니다 ACTION.

명령:

```
aws cognito-idp set-risk-configuration --user-pool-id us-west-2_aaaaaaaaa  --
compromised-credentials-risk-
configuration EventFilter=SIGN_UP,Actions={EventAction=NO_ACTION}
```

출력:

```
{
  "RiskConfiguration": {
      "UserPoolId": "us-west-2_aaaaaaaaa",
      "CompromisedCredentialsRiskConfiguration": {
          "EventFilter": [
              "SIGN_UP"
          ],
          "Actions": {
              "EventAction": "NO_ACTION"
          }
      }
  }
}
```

- 자세한 API 내용은 명령 참조SetRiskConfiguration의 섹션을 참조하세요. AWS CLI

### `set-ui-customization`

다음 코드 예시에서는 `set-ui-customization`을 사용하는 방법을 보여 줍니다.

AWS CLI

UI 사용자 지정을 설정하려면

이 예제에서는 사용자 풀의 CSS 설정을 사용자 지정합니다.

명령:

```
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"
```

출력:

```
{
  "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"
  }
}
```

- 자세한 API 내용은 명령 참조SetUiCustomization의 섹션을 참조하세요. AWS CLI

### `set-user-mfa-preference`

다음 코드 예시에서는 `set-user-mfa-preference`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 MFA 설정을 설정하려면

다음 `set-user-mfa-preference` 예제에서는 MFA 전송 옵션을 수정합니다. MFA 전송 미디어를 로 변경합니다SMS.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Cognito 개발자 안내서의 사용자 풀MFA에 추가를 참조하세요.

- 자세한 API 내용은 명령 참조SetUserMfaPreference의 섹션을 참조하세요. AWS CLI

### `set-user-settings`

다음 코드 예시에서는 `set-user-settings`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 설정을 설정하려면

이 예제에서는 MFA 전송 기본 설정을 로 설정합니다EMAIL.

명령:

```
aws cognito-idp set-user-settings --access-token ACCESS_TOKEN --mfa-
options DeliveryMedium=EMAIL
```

- 자세한 API 내용은 명령 참조SetUserSettings의 섹션을 참조하세요. AWS CLI

### sign-up

다음 코드 예시에서는 sign-up을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 가입

이 예시에서는 jane@example.com에 가입합니다.

명령:

```
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"
```

출력:

```
{
  "UserConfirmed": false,
  "UserSub": "e04d60a6-45dc-441c-a40b-e25a787d4862"
}
```

- 자세한 API 내용은 명령 참조SignUp의 섹션을 참조하세요. AWS CLI

### start-user-import-job

다음 코드 예시에서는 start-user-import-job을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 가져오기 작업을 시작하려면

이 예제에서는 사용자 입력 작업을 시작합니다.

사용자 가져오기에 대한 자세한 내용은 CSV 파일에서 사용자 풀로 사용자 가져오기를 참조하세요.

명령:

```
aws cognito-idp start-user-import-job --user-pool-id us-west-2_aaaaaaaaa --job-
id import-TZqNQvDRnW
```

출력:

```
{
  "UserImportJob": {
      "JobName": "import-Test10",
      "JobId": "import-lmpxSOuIzH",
      "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
  }
}
```

- 자세한 API 내용은 명령 참조StartUserImportJob의 섹션을 참조하세요. AWS CLI

### stop-user-import-job

다음 코드 예시에서는 stop-user-import-job을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 가져오기 작업을 중지하려면

이 예제는 사용자 입력 작업을 중지합니다.

사용자 가져오기에 대한 자세한 내용은 CSV 파일에서 사용자 풀로 사용자 가져오기를 참조하세요.

명령:

```
aws cognito-idp stop-user-import-job --user-pool-id us-west-2_aaaaaaaaa --job-
id import-TZqNQvDRnW
```

출력:

```
{
  "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."
  }
}
```

- 자세한 API 내용은 명령 참조StopUserImportJob의 섹션을 참조하세요. AWS CLI

### update-auth-event-feedback

다음 코드 예시에서는 update-auth-event-feedback을 사용하는 방법을 보여 줍니다.

AWS CLI

인증 이벤트 피드백을 업데이트하려면

이 예제에서는 권한 부여 이벤트 피드백을 업데이트합니다. 이벤트 “Valid”를 표시합니다.

명령:

```
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"
```

- 자세한 API 내용은 명령 참조UpdateAuthEventFeedback의 섹션을 참조하세요. AWS CLI

### update-device-status

다음 코드 예시에서는 update-device-status을 사용하는 방법을 보여 줍니다.

AWS CLI

디바이스 상태를 업데이트하려면

이 예제에서는 디바이스의 상태를 “not_remembered”로 업데이트합니다.

명령:

```
aws cognito-idp update-device-status --access-token ACCESS_TOKEN --device-
key DEVICE_KEY --device-remembered-status "not_remembered"
```

- 자세한 API 내용은 명령 참조UpdateDeviceStatus의 섹션을 참조하세요. AWS CLI

### update-group

다음 코드 예시에서는 update-group을 사용하는 방법을 보여 줍니다.

AWS CLI

그룹을 업데이트하려면

이 예제에서는 에 대한 설명과 우선 순위를 업데이트합니다 MyGroup.

명령:

```
aws cognito-idp update-group --user-pool-id us-west-2_aaaaaaaaa --group-name MyGroup
 --description "New description" --precedence 2
```

출력:

```
{
```

```
  "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
  }
}
```

- 자세한 API 내용은 명령 참조UpdateGroup의 섹션을 참조하세요. AWS CLI

### update-resource-server

다음 코드 예시에서는 update-resource-server을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 서버를 업데이트하려면

이 예제에서는 리소스 서버 날씨를 업데이트합니다. 새 범위가 추가됩니다.

명령:

```
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"
```

출력:

```
{
  "ResourceServer": {
      "UserPoolId": "us-west-2_aaaaaaaaa",
      "Identifier": "weather.example.com",
      "Name": "Happy",
      "Scopes": [
          {
              "ScopeName": "NewScope",
              "ScopeDescription": "New scope description"
          }
      ]
  }
```

```
}
```

- 자세한 API 내용은 명령 참조UpdateResourceServer의 섹션을 참조하세요. AWS CLI

### update-user-attributes

다음 코드 예시에서는 update-user-attributes을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 속성을 업데이트하려면

이 예제에서는 사용자 속성 “nickname”을 업데이트합니다.

명령:

```
aws cognito-idp update-user-attributes --access-token ACCESS_TOKEN --user-attributes
 Name="nickname",Value="Dan"
```

- 자세한 API 내용은 명령 참조UpdateUserAttributes의 섹션을 참조하세요. AWS CLI

### update-user-pool-client

다음 코드 예시에서는 update-user-pool-client을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 풀 클라이언트를 업데이트하려면

이 예제에서는 사용자 풀 클라이언트의 이름을 업데이트합니다. 또한 쓰기 가능한 속성 “nickname”도 추가합니다.

명령:

```
aws cognito-idp update-user-pool-client --user-pool-id us-west-2_aaaaaaaaa --
client-id 3n4b5urk1ft4fl3mg5e62d9ado --client-name "NewClientName" --write-
attributes "nickname"
```

출력:

```
{
  "UserPoolClient": {
```

```
        "UserPoolId": "us-west-2_aaaaaaaaa",
        "ClientName": "NewClientName",
        "ClientId": "3n4b5urk1ft4fl3mg5e62d9ado",
        "LastModifiedDate": 1548802761.334,
        "CreationDate": 1548178931.258,
        "RefreshTokenValidity": 30,
        "WriteAttributes": [
            "nickname"
        ],
        "AllowedOAuthFlowsUserPoolClient": false
    }
}
```

- 자세한 API 내용은 명령 참조UpdateUserPoolClient의 섹션을 참조하세요. AWS CLI

### update-user-pool

다음 코드 예시에서는 update-user-pool을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 풀을 업데이트하려면

다음 update-user-pool 예제는 사용 가능한 각 구성 옵션에 대한 예제 구문으로 사용자 풀을 수정합니다. 사용자 풀을 업데이트하려면 이전에 구성된 모든 옵션을 지정해야 합니다. 그렇지 않으면 옵션이 기본값으로 재설정됩니다.

```
aws cognito-idp update-user-pool --user-pool-id us-west-2_EXAMPLE \
    --policies PasswordPolicy=
\{MinimumLength=6,RequireUppercase=true,RequireLowercase=true,RequireNumbers=true,RequireSym
 \
    --deletion-protection ACTIVE \
    --lambda-config PreSignUp="arn:aws:lambda:us-
west-2:123456789012:function:cognito-test-presignup-
function",PreTokenGeneration="arn:aws:lambda:us-
west-2:123456789012:function:cognito-test-pretoken-function" \
    --auto-verified-attributes "phone_number" "email" \
    --verification-message-template \{\"SmsMessage\":\""Your code is
 {####}"\",\"EmailMessage\":\""Your code is {####}"\",\"EmailSubject\":\""Your
 verification code"\",\"EmailMessageByLink\":\""Click {##here##} to verify
 your email address."\",\"EmailSubjectByLink\":\""Your verification link"\",
\"DefaultEmailOption\":\"CONFIRM_WITH_LINK\"\} \
    --sms-authentication-message "Your code is {####}" \
```

```
    --user-attribute-update-settings
 AttributesRequireVerificationBeforeUpdate="email","phone_number" \
    --mfa-configuration "OPTIONAL" \
    --device-
configuration ChallengeRequiredOnNewDevice=true,DeviceOnlyRememberedOnUserPrompt=true
 \
    --email-configuration SourceArn="arn:aws:ses:us-
west-2:123456789012:identity/admin@example.com",ReplyToEmailAddress="amdin
+noreply@example.com",EmailSendingAccount=DEVELOPER,From="admin@amazon.com",ConfigurationSet
configuration-set" \
    --sms-configuration SnsCallerArn="arn:aws:iam::123456789012:role/service-role/
SNS-SMS-Role",ExternalId="12345",SnsRegion="us-west-2" \
    --admin-create-user-config AllowAdminCreateUserOnly=false,InviteMessageTemplate=
\{SMSMessage=\""Welcome {username}. Your confirmation code is
 {####}"\",EmailMessage=\""Welcome {username}. Your confirmation code is
 {####}"\",EmailSubject=\""Welcome to MyMobileGame"\"\} \
    --user-pool-tags "Function"="MyMobileGame","Developers"="Berlin" \
    --admin-create-user-config AllowAdminCreateUserOnly=false,InviteMessageTemplate=
\{SMSMessage=\""Welcome {username}. Your confirmation code is
 {####}"\",EmailMessage=\""Welcome {username}. Your confirmation code is
 {####}"\",EmailSubject=\""Welcome to MyMobileGame"\"\} \
    --user-pool-add-ons AdvancedSecurityMode="AUDIT" \
    --account-recovery-setting RecoveryMechanisms=
\[\{Priority=1,Name="verified_email"\},\{Priority=2,Name="verified_phone_number"\}\]
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Cognito 개발자 안내서의 사용자 풀 구성 업데이트를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateUserPool의 섹션을 참조하세요. AWS CLI

## 를 사용한 Amazon Comprehend 예제 AWS CLI

다음 코드 예제에서는 Amazon Comprehend 와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### batch-detect-dominant-language

다음 코드 예시에서는 batch-detect-dominant-language을 사용하는 방법을 보여 줍니다.

AWS CLI

여러 입력 텍스트의 지배적 언어를 감지하려면

다음 batch-detect-dominant-language 예제에서는 여러 입력 텍스트를 분석하고 각 의 지배적 언어를 반환합니다. 사전 훈련된 모델 신뢰도 점수도 각 예측에 대해 출력됩니다.

```
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."
```

출력:

```
{
    "ResultList": [
        {
            "Index": 0,
            "Languages": [
                {
                    "LanguageCode": "en",
                    "Score": 0.9986501932144165
                }
            ]
        }
    ],
    "ErrorList": []
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 주로 사용되는 언어를 참조하세요.

- 자세한 API 내용은 명령 참조BatchDetectDominantLanguage의 섹션을 참조하세요. AWS CLI

### batch-detect-entities

다음 코드 예시에서는 batch-detect-entities을 사용하는 방법을 보여 줍니다.

AWS CLI

여러 입력 텍스트에서 엔터티를 감지하려면

다음 batch-detect-entities 예제에서는 여러 입력 텍스트를 분석하고 각 의 명명된 엔터티를 반환합니다. 사전 훈련된 모델의 신뢰도 점수도 각 예측에 대해 출력됩니다.

```
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."
```

출력:

```
{
    "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": []
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 엔티티를 참조하세요.

- 자세한 API 내용은 명령 참조BatchDetectEntities의 섹션을 참조하세요. AWS CLI

### batch-detect-key-phrases

다음 코드 예시에서는 batch-detect-key-phrases을 사용하는 방법을 보여 줍니다.

AWS CLI

여러 텍스트 입력의 키 구문을 감지하려면

다음 batch-detect-key-phrases 예제에서는 여러 입력 텍스트를 분석하고 각 의 키 명사 문구를 반환합니다. 각 예측에 대한 사전 훈련된 모델의 신뢰도 점수도 출력됩니다.

```
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."
```

출력:

```
{
    "ResultList": [
        {
            "Index": 0,
            "KeyPhrases": [
                {
                    "Score": 0.99700927734375,
                    "Text": "Zhang Wei",
                    "BeginOffset": 6,
                    "EndOffset": 15
                },
                {
                    "Score": 0.9929308891296387,
                    "Text": "John",
                    "BeginOffset": 22,
                    "EndOffset": 26
                },
                {
                    "Score": 0.9997230172157288,
                    "Text": "the trip",
                    "BeginOffset": 49,
                    "EndOffset": 57
                },
                {
                    "Score": 0.9999470114707947,
                    "Text": "next Saturday",
                    "BeginOffset": 62,
                    "EndOffset": 75
                }
            ]
        },
        {
            "Index": 1,
            "KeyPhrases": [
                {
                    "Score": 0.8358274102210999,
                    "Text": "Dear Jane",
                    "BeginOffset": 0,
                    "EndOffset": 9
                },
                {
                    "Score": 0.989359974861145,
                    "Text": "Your AnyCompany Financial Services",
```

```
                    "BeginOffset": 11,
                    "EndOffset": 45
                },
                {
                    "Score": 0.8812323808670044,
                    "Text": "LLC credit card account 1111-XXXX-1111-XXXX",
                    "BeginOffset": 47,
                    "EndOffset": 90
                },
                {
                    "Score": 0.9999381899833679,
                    "Text": "a minimum payment",
                    "BeginOffset": 95,
                    "EndOffset": 112
                },
                {
                    "Score": 0.9997439980506897,
                    "Text": ".53",
                    "BeginOffset": 116,
                    "EndOffset": 119
                },
                {
                    "Score": 0.996875524520874,
                    "Text": "July 31st",
                    "BeginOffset": 135,
                    "EndOffset": 144
                }
            ]
        },
        {
            "Index": 2,
            "KeyPhrases": [
                {
                    "Score": 0.9990295767784119,
                    "Text": "customer feedback",
                    "BeginOffset": 12,
                    "EndOffset": 29
                },
                {
                    "Score": 0.9994127750396729,
                    "Text": "Sunshine Spa",
                    "BeginOffset": 33,
                    "EndOffset": 45
                },
```

```
                {
                    "Score": 0.9892991185188293,
                    "Text": "123 Main St",
                    "BeginOffset": 47,
                    "EndOffset": 58
                },
                {
                    "Score": 0.9969810843467712,
                    "Text": "Alice",
                    "BeginOffset": 75,
                    "EndOffset": 80
                },
                {
                    "Score": 0.9703696370124817,
                    "Text": "AnySpa@example.com",
                    "BeginOffset": 84,
                    "EndOffset": 99
                }
            ]
        }
    ],
    "ErrorList": []
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 핵심 문구를 참조하세요.

- 자세한 API 내용은 명령 참조BatchDetectKeyPhrases의 섹션을 참조하세요. AWS CLI

### batch-detect-sentiment

다음 코드 예시에서는 batch-detect-sentiment을 사용하는 방법을 보여 줍니다.

AWS CLI

여러 입력 텍스트의 일반적인 감정을 감지하려면

다음 batch-detect-sentiment 예제에서는 여러 입력 텍스트를 분석하고 일반적인 감정(각각의 POSITIVE, NEGATIVE, NEUTRAL MIXED또는 )을 반환합니다.

```
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
```

출력:

```
{
    "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": []
}
```

자세한 내용은 Amazon Comprehend 개발자 안내서의 감정을 참조하세요.

- 자세한 API 내용은 명령 참조BatchDetectSentiment의 섹션을 참조하세요. AWS CLI

### batch-detect-syntax

다음 코드 예시에서는 batch-detect-syntax을 사용하는 방법을 보여 줍니다.

AWS CLI

여러 입력 텍스트에서 단어의 구문과 스피치 부분을 검사하려면

다음 batch-detect-syntax 예제에서는 여러 입력 텍스트의 구문을 분석하고 음성의 다양한 부분을 반환합니다. 사전 훈련된 모델의 신뢰도 점수도 각 예측에 대해 출력됩니다.

```
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
```

출력:

```
{
    "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": []
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 구문 분석을 참조하세요.

- 자세한 API 내용은 명령 참조BatchDetectSyntax의 섹션을 참조하세요. AWS CLI

### `batch-detect-targeted-sentiment`

다음 코드 예시에서는 `batch-detect-targeted-sentiment`을 사용하는 방법을 보여 줍니다.

AWS CLI

여러 입력 텍스트에 대해 감성과 이름이 지정된 각 엔터티를 감지하려면

다음 batch-detect-targeted-sentiment 예제에서는 여러 입력 텍스트를 분석하고 각 엔터티에 연결된 일반적인 감성과 함께 명명된 엔터티를 반환합니다. 사전 훈련된 모델의 신뢰도 점수도 각 예측에 대해 출력됩니다.

```
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."
```

출력:

```
{
    "ResultList": [
        {
            "Index": 0,
            "Entities": [
                {
                    "DescriptiveMentionIndex": [
                        0
                    ],
                    "Mentions": [
                        {
                            "Score": 0.9999009966850281,
                            "GroupScore": 1.0,
                            "Text": "movie",
                            "Type": "MOVIE",
                            "MentionSentiment": {
                                "Sentiment": "NEGATIVE",
                                "SentimentScore": {
                                    "Positive": 0.13887299597263336,
                                    "Negative": 0.8057460188865662,
                                    "Neutral": 0.05525200068950653,
                                    "Mixed": 0.00012799999967683107
                                }
                            },
                            "BeginOffset": 5,
                            "EndOffset": 10
                        }
                    ]
                },
                {
                    "DescriptiveMentionIndex": [
                        0
```

```
                ],
                "Mentions": [
                    {
                        "Score": 0.9921110272407532,
                        "GroupScore": 1.0,
                        "Text": "original",
                        "Type": "MOVIE",
                        "MentionSentiment": {
                            "Sentiment": "POSITIVE",
                            "SentimentScore": {
                                "Positive": 0.9999989867210388,
                                "Negative": 9.999999974752427e-07,
                                "Neutral": 0.0,
                                "Mixed": 0.0
                            }
                        },
                        "BeginOffset": 34,
                        "EndOffset": 42
                    }
                ]
            }
        ]
    },
    {
        "Index": 1,
        "Entities": [
            {
                "DescriptiveMentionIndex": [
                    0
                ],
                "Mentions": [
                    {
                        "Score": 0.7545599937438965,
                        "GroupScore": 1.0,
                        "Text": "trail",
                        "Type": "OTHER",
                        "MentionSentiment": {
                            "Sentiment": "POSITIVE",
                            "SentimentScore": {
                                "Positive": 1.0,
                                "Negative": 0.0,
                                "Neutral": 0.0,
                                "Mixed": 0.0
                            }
```

```
                        },
                        "BeginOffset": 4,
                        "EndOffset": 9
                    }
                ]
            },
            {
                "DescriptiveMentionIndex": [
                    0
                ],
                "Mentions": [
                    {
                        "Score": 0.9999960064888,
                        "GroupScore": 1.0,
                        "Text": "today",
                        "Type": "DATE",
                        "MentionSentiment": {
                            "Sentiment": "NEUTRAL",
                            "SentimentScore": {
                                "Positive": 9.000000318337698e-06,
                                "Negative": 1.9999999949504854e-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": []
```

```
}
```

자세한 내용은 Amazon Comprehend 개발자 안내서의 대상 감정을 참조하세요.

- 자세한 API 내용은 명령 참조BatchDetectTargetedSentiment의 섹션을 참조하세요. AWS CLI

### `classify-document`

다음 코드 예시에서는 `classify-document`을 사용하는 방법을 보여 줍니다.

AWS CLI

모델별 엔드포인트로 문서를 분류하려면

다음 `classify-document` 예제에서는 사용자 지정 모델의 엔드포인트가 있는 문서를 분류합니다. 이 예제의 모델은 스팸 또는 비스팸 또는 'ham'으로 레이블이 지정된 sms 메시지가 포함된 데이터 세트에서 훈련되었습니다.

```
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"
```

출력:

```
{
    "Classes": [
        {
            "Name": "spam",
            "Score": 0.9998599290847778
        },
        {
            "Name": "ham",
            "Score": 0.00014001205272506922
        }
    ]
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 사용자 지정 분류를 참조하세요.

- 자세한 API 내용은 명령 참조ClassifyDocument의 섹션을 참조하세요. AWS CLI

### contains-pii-entities

다음 코드 예시에서는 contains-pii-entities을 사용하는 방법을 보여 줍니다.

AWS CLI

입력 텍스트의 PII 정보 존재를 분석하려면

다음 contains-pii-entities 예제에서는 입력 텍스트에 개인 식별 정보(PII)가 있는지 분석하고 이름, 주소, 은행 계좌 번호 또는 전화번호와 같이 식별된 PII 엔터티 유형의 레이블을 반환합니다.

```
aws comprehend contains-pii-entities \
    --language-code en \
    --text "Hello Zhang Wei, I am John. Your AnyCompany Financial Services, LLC
 credit card
       account 1111-XXXX-1111-XXXX has a minimum payment of $24.53 that is due by
 July 31st. Based on your autopay settings,
       we will withdraw your payment on the due date from your bank account number
 XXXXXX1111 with the routing number XXXXX0000.
       Customer feedback for Sunshine Spa, 100 Main St, Anywhere. Send comments to
 Alice at AnySpa@example.com."
```

출력:

```
{
    "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
        }
}
```

자세한 내용은 Amazon Comprehend 개발자 안내서의 개인 식별 정보(PII)를 참조하세요.

- 자세한 API 내용은 명령 참조ContainsPiiEntities의 섹션을 참조하세요. AWS CLI

### `create-dataset`

다음 코드 예시에서는 `create-dataset`을 사용하는 방법을 보여 줍니다.

AWS CLI

플라이휠 데이터 세트를 생성하려면

다음 `create-dataset` 예제에서는 플라이휠에 대한 데이터 세트를 생성합니다. 이 데이터 세트는 `--dataset-type` 태그에 지정된 추가 훈련 데이터로 사용됩니다.

```
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
```

`file://inputConfig.json`의 콘텐츠:

```
{
    "DataFormat": "COMPREHEND_CSV",
    "DocumentClassifierInputDataConfig": {
        "S3Uri": "s3://DOC-EXAMPLE-BUCKET/training-data.csv"
    }
}
```

출력:

```
{
    "DatasetArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/flywheel-
entity/dataset/example-dataset"
```

```
}
```

자세한 내용은 Amazon Comprehend 개발자 안내서의 Flywheel 개요를 참조하세요.

- 자세한 API 내용은 명령 참조CreateDataset의 섹션을 참조하세요. AWS CLI

### `create-document-classifier`

다음 코드 예시에서는 `create-document-classifier`을 사용하는 방법을 보여 줍니다.

AWS CLI

문서 분류자를 만들어 문서 분류

다음 `create-document-classifier` 예제에서는 문서 분류자 모델의 학습 프로세스를 시작합니다. 교육 데이터 파일 `training.csv`는 `--input-data-config` 태그에 있습니다. `training.csv`는 첫 번째 열에 레이블 또는 분류가 제공되고 두 번째 열에 문서가 제공되는 2열 문서입니다.

```
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
```

출력:

```
{
    "DocumentClassifierArn": "arn:aws:comprehend:us-west-2:111122223333:document-
classifier/example-classifier"
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 사용자 지정 분류를 참조하세요.

- 자세한 API 내용은 명령 참조CreateDocumentClassifier의 섹션을 참조하세요. AWS CLI

### `create-endpoint`

다음 코드 예시에서는 `create-endpoint`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 모델의 엔드포인트를 생성하려면

다음 create-endpoint 예제에서는 이전에 훈련된 사용자 지정 모델에 대한 동기 추론을 위한 엔드포인트를 생성합니다.

```
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
```

출력:

```
{
    "EndpointArn": "arn:aws:comprehend:us-west-2:111122223333:document-classifier-
endpoint/example-classifier-endpoint-1"
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 Amazon Comprehend 엔드포인트 관리를 참조하세요.

- 자세한 API 내용은 명령 참조CreateEndpoint의 섹션을 참조하세요. AWS CLI

### create-entity-recognizer

다음 코드 예시에서는 create-entity-recognizer을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 엔터티 인식기를 생성하려면

다음 create-entity-recognizer 예제에서는 사용자 지정 개체 인식기 모델에 대한 훈련 프로세스를 시작합니다. 이 예제에서는 훈련 문서, raw_text.csv및 CSV 개체 목록이 포함된 CSV 파일을 사용하여 모델을 entity_list.csv 훈련합니다. entity-list.csv에는 텍스트 및 유형 열이 포함되어 있습니다.

```
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
```

출력:

```
{
    "EntityRecognizerArn": "arn:aws:comprehend:us-west-2:111122223333:example-
entity-recognizer/entityrecognizer1"
}
```

자세한 내용은 Amazon Comprehend 개발자 안내서의 사용자 지정 엔터티 인식을 참조하세요.

- 자세한 API 내용은 명령 참조CreateEntityRecognizer의 섹션을 참조하세요. AWS CLI

### `create-flywheel`

다음 코드 예시에서는 `create-flywheel`을 사용하는 방법을 보여 줍니다.

AWS CLI

플라이휠을 생성하려면

다음 `create-flywheel` 예제에서는 문서 분류 또는 엔터티 인식 모델의 지속적인 훈련을 오케스트레이션하기 위한 플라이휠을 생성합니다. 이 예제의 플라이휠은 `--active-model-arn` 태그로 지정된 기존 훈련된 모델을 관리하도록 생성됩니다. 플라이휠이 생성되면 `--input-data-lake` 태그에 데이터 레이크가 생성됩니다.

```
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"
```

출력:

```
{
    "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/example-
flywheel"
```

```
}
```

자세한 내용은 Amazon Comprehend 개발자 안내서의 플라이휠 개요를 참조하세요.

- 자세한 API 내용은 명령 참조CreateFlywheel의 섹션을 참조하세요. AWS CLI

### delete-document-classifier

다음 코드 예시에서는 delete-document-classifier을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 문서 분류자 삭제

다음 delete-document-classifier 예제에서는 사용자 지정 문서 분류자 모델을 삭제합니다.

```
aws comprehend delete-document-classifier \
    --document-classifier-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/example-classifier-1
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Comprehend 개발자 가이드의 Amazon Comprehend 엔드포인트 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteDocumentClassifier의 섹션을 참조하세요. AWS CLI

### delete-endpoint

다음 코드 예시에서는 delete-endpoint을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 모델의 엔드포인트를 삭제하려면

다음 delete-endpoint 예제에서는 모델별 엔드포인트를 삭제합니다. 모델을 삭제하려면 모든 엔드포인트를 삭제해야 합니다.

```
aws comprehend delete-endpoint \
    --endpoint-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier-endpoint/example-classifier-endpoint-1
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Comprehend 개발자 가이드의 Amazon Comprehend 엔드포인트 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteEndpoint의 섹션을 참조하세요. AWS CLI

### `delete-entity-recognizer`

다음 코드 예시에서는 `delete-entity-recognizer`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 엔터티 인식기 모델을 삭제하려면

다음 `delete-entity-recognizer` 예제에서는 사용자 지정 엔터티 인식기 모델을 삭제합니다.

```
aws comprehend delete-entity-recognizer \
    --entity-recognizer-arn arn:aws:comprehend:us-west-2:111122223333:entity-recognizer/example-entity-recognizer-1
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Comprehend 개발자 가이드의 Amazon Comprehend 엔드포인트 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteEntityRecognizer의 섹션을 참조하세요. AWS CLI

### `delete-flywheel`

다음 코드 예시에서는 `delete-flywheel`을 사용하는 방법을 보여 줍니다.

AWS CLI

플라이휠을 삭제하려면

다음 `delete-flywheel` 예제에서는 플라이휠을 삭제합니다. 플라이휠과 연결된 데이터 레이크 또는 모델은 삭제되지 않습니다.

```
aws comprehend delete-flywheel \
    --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/example-flywheel-1
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Comprehend 개발자 안내서의 Flywheel 개요를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteFlywheel의 섹션을 참조하세요. AWS CLI

### delete-resource-policy

다음 코드 예시에서는 delete-resource-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 기반 정책을 삭제하려면

다음 delete-resource-policy 예제에서는 Amazon Comprehend 리소스에서 리소스 기반 정책을 삭제합니다.

```
aws comprehend delete-resource-policy \
    --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/example-classifier-1/version/1
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Comprehend 개발자 안내서의 AWS 계정 간에 사용자 지정 모델 복사를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteResourcePolicy의 섹션을 참조하세요. AWS CLI

### describe-dataset

다음 코드 예시에서는 describe-dataset을 사용하는 방법을 보여 줍니다.

AWS CLI

플라이휠 데이터 세트를 설명하려면

다음 describe-dataset 예제에서는 플라이휠 데이터 세트의 속성을 가져옵니다.

```
aws comprehend describe-dataset \
    --dataset-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/flywheel-entity/dataset/example-dataset
```

출력:

```
{
```

```
    "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"
    }
}
```

자세한 내용은 Amazon Comprehend 개발자 안내서의 플라이휠 개요를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeDataset의 섹션을 참조하세요. AWS CLI

### `describe-document-classification-job`

다음 코드 예시에서는 `describe-document-classification-job`을 사용하는 방법을 보여 줍니다.

AWS CLI

문서 분류 작업 설명

다음 `describe-document-classification-job` 예제는 비동기 문서 분류 작업의 속성을 가져옵니다.

```
aws comprehend describe-document-classification-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 사용자 지정 분류를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeDocumentClassificationJob의 섹션을 참조하세요. AWS CLI

### `describe-document-classifier`

다음 코드 예시에서는 `describe-document-classifier`을 사용하는 방법을 보여 줍니다.

AWS CLI

문서 분류기 설명

다음 `describe-document-classifier` 예제에서는 사용자 지정 문서 분류자 모델을 삭제합니다.

```
aws comprehend describe-document-classifier \
    --document-classifier-arn arn:aws:comprehend:us-west-2:111122223333:document-
classifier/example-classifier-1
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 사용자 지정 모델 생성 및 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeDocumentClassifier의 섹션을 참조하세요. AWS CLI

### `describe-dominant-language-detection-job`

다음 코드 예시에서는 `describe-dominant-language-detection-job`을 사용하는 방법을 보여줍니다.

AWS CLI

지배적인 언어 감지 감지 작업을 설명합니다.

다음 describe-dominant-language-detection-job 예제에서는 비동기 우성 언어 감지 작업의 속성을 가져옵니다.

```
aws comprehend describe-dominant-language-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 Amazon Comprehend 통찰력 비동기 분석을 참조하십시오.

- 자세한 API 내용은 명령 참조DescribeDominantLanguageDetectionJob의 섹션을 참조하세요. AWS CLI

### describe-endpoint

다음 코드 예시에서는 describe-endpoint을 사용하는 방법을 보여 줍니다.

AWS CLI

특정 엔드포인트를 설명하려면

다음 describe-endpoint 예제에서는 모델별 엔드포인트의 속성을 가져옵니다.

```
aws comprehend describe-endpoint \
    --endpoint-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier-endpoint/example-classifier-endpoint
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 Amazon Comprehend 엔드포인트 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeEndpoint의 섹션을 참조하세요. AWS CLI

### describe-entities-detection-job

다음 코드 예시에서는 describe-entities-detection-job을 사용하는 방법을 보여 줍니다.

AWS CLI

엔터티 감지 작업을 설명하려면

다음 describe-entities-detection-job 예제에서는 비동기 엔터티 감지 작업의 속성을 가져옵니다.

```
aws comprehend describe-entities-detection-job \
```

```
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 Amazon Comprehend 통찰력 비동기 분석을 참조하십시오.

- 자세한 API 내용은 명령 참조DescribeEntitiesDetectionJob의 섹션을 참조하세요. AWS CLI

### `describe-entity-recognizer`

다음 코드 예시에서는 `describe-entity-recognizer`을 사용하는 방법을 보여 줍니다.

AWS CLI

엔터티 인식기를 설명하려면

다음 `describe-entity-recognizer` 예제에서는 사용자 지정 엔터티 인식기 모델의 속성을 가져옵니다.

```
aws comprehend describe-entity-recognizer \
    entity-recognizer-arn arn:aws:comprehend:us-west-2:111122223333:entity-
recognizer/business-recongizer-1/version/1
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Comprehend 개발자 안내서의 사용자 지정 엔터티 인식을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeEntityRecognizer의 섹션을 참조하세요. AWS CLI

### describe-events-detection-job

다음 코드 예시에서는 describe-events-detection-job을 사용하는 방법을 보여 줍니다.

AWS CLI

이벤트 감지 작업을 설명합니다.

다음 describe-events-detection-job 예제에서는 비동기 이벤트 감지 작업의 속성을 가져옵니다.

```
aws comprehend describe-events-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

출력:

```
{
    "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"
        ]
    }
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 Amazon Comprehend 통찰력 비동기 분석을 참조하십시오.

- 자세한 API 내용은 명령 참조DescribeEventsDetectionJob의 섹션을 참조하세요. AWS CLI

### `describe-flywheel-iteration`

다음 코드 예시에서는 `describe-flywheel-iteration`을 사용하는 방법을 보여 줍니다.

AWS CLI

플라이휠 반복을 설명하려면

다음 `describe-flywheel-iteration` 예제에서는 플라이휠 반복의 속성을 가져옵니다.

```
aws comprehend describe-flywheel-iteration \
    --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/example-
flywheel \
    --flywheel-iteration-id 20232222AEXAMPLE
```

출력:

```
{
    "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/"
    }
}
```

자세한 내용은 Amazon Comprehend 개발자 안내서의 Flywheel 개요를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeFlywheelIteration의 섹션을 참조하세요. AWS CLI

### `describe-flywheel`

다음 코드 예시에서는 `describe-flywheel`을 사용하는 방법을 보여 줍니다.

AWS CLI

플라이휠을 설명하려면

다음 describe-flywheel 예제에서는 플라이 휠의 속성을 가져옵니다. 이 예제에서 플라이휠과 연결된 모델은 문서를 스팸 또는 스팸 아님 또는 'ham'으로 분류하도록 훈련된 사용자 지정 분류기 모델입니다.

```
aws comprehend describe-flywheel \
    --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/example-flywheel
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Comprehend 개발자 안내서의 플라이휠 개요를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeFlywheel의 섹션을 참조하세요. AWS CLI

### describe-key-phrases-detection-job

다음 코드 예시에서는 describe-key-phrases-detection-job을 사용하는 방법을 보여 줍니다.

AWS CLI

키 문구 감지 작업을 설명하려면

다음 describe-key-phrases-detection-job 예제에서는 비동기식 키 구문 감지 작업의 속성을 가져옵니다.

```
aws comprehend describe-key-phrases-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 Amazon Comprehend 통찰력 비동기 분석을 참조하십시오.

- 자세한 API 내용은 명령 참조DescribeKeyPhrasesDetectionJob의 섹션을 참조하세요. AWS CLI

### describe-pii-entities-detection-job

다음 코드 예시에서는 describe-pii-entities-detection-job을 사용하는 방법을 보여 줍니다.

AWS CLI

PII 개체 감지 작업을 설명하려면

다음 describe-pii-entities-detection-job 예제에서는 비동기 pii 엔터티 감지 작업의 속성을 가져옵니다.

```
aws comprehend describe-pii-entities-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 Amazon Comprehend 통찰력 비동기 분석을 참조하십시오.

- 자세한 API 내용은 명령 참조DescribePiiEntitiesDetectionJob의 섹션을 참조하세요. AWS CLI

### describe-resource-policy

다음 코드 예시에서는 describe-resource-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

모델에 연결된 리소스 정책을 설명하려면

다음 describe-resource-policy 예제에서는 모델에 연결된 리소스 기반 정책의 속성을 가져옵니다.

```
aws comprehend describe-resource-policy \
    --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/
example-classifier/version/1
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon Comprehend 개발자 안내서의 AWS 계정 간에 사용자 지정 모델 복사를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeResourcePolicy의 섹션을 참조하세요. AWS CLI

### describe-sentiment-detection-job

다음 코드 예시에서는 describe-sentiment-detection-job을 사용하는 방법을 보여 줍니다.

AWS CLI

감정 감지 작업을 설명하려면

다음 describe-sentiment-detection-job 예제에서는 비동기 감정 감지 작업의 속성을 가져옵니다.

```
aws comprehend describe-sentiment-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 Amazon Comprehend 통찰력 비동기 분석을 참조하십시오.

- 자세한 API 내용은 명령 참조DescribeSentimentDetectionJob의 섹션을 참조하세요. AWS CLI

### describe-targeted-sentiment-detection-job

다음 코드 예시에서는 describe-targeted-sentiment-detection-job을 사용하는 방법을 보여 줍니다.

AWS CLI

대상 감정 감지 작업을 설명하려면

다음 `describe-targeted-sentiment-detection-job` 예제에서는 비동기식 대상 감정 감지 작업의 속성을 가져옵니다.

```
aws comprehend describe-targeted-sentiment-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 Amazon Comprehend 통찰력 비동기 분석을 참조하십시오.

- 자세한 API 내용은 명령 참조DescribeTargetedSentimentDetectionJob의 섹션을 참조하세요. AWS CLI

### describe-topics-detection-job

다음 코드 예시에서는 describe-topics-detection-job을 사용하는 방법을 보여 줍니다.

AWS CLI

주제 탐지 작업 설명

다음 describe-topics-detection-job 예제는 비동기 주제 탐지 작업의 속성을 가져옵니다.

```
aws comprehend describe-topics-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 Amazon Comprehend 통찰력 비동기 분석을 참조하십시오.

- 자세한 API 내용은 명령 참조DescribeTopicsDetectionJob의 섹션을 참조하세요. AWS CLI

### detect-dominant-language

다음 코드 예시에서는 detect-dominant-language을 사용하는 방법을 보여 줍니다.

AWS CLI

입력 텍스트에서 주로 사용되는 언어 탐지

다음 detect-dominant-language은(는) 입력 텍스트를 분석하고 주로 사용되는 언어를 식별합니다. 사전 훈련된 모델의 신뢰도 점수도 출력됩니다.

```
aws comprehend detect-dominant-language \
    --text "It is a beautiful day in Seattle."
```

출력:

```
{
    "Languages": [
        {
            "LanguageCode": "en",
            "Score": 0.9877256155014038
        }
    ]
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 주로 사용되는 언어를 참조하세요.

- 자세한 API 내용은 명령 참조DetectDominantLanguage의 섹션을 참조하세요. AWS CLI

### detect-entities

다음 코드 예시에서는 detect-entities을 사용하는 방법을 보여 줍니다.

AWS CLI

입력 텍스트에서 명명된 엔터티를 감지하려면

다음 detect-entities 예제에서는 입력 텍스트를 분석하고 이름이 지정된 엔티티를 반환합니다. 사전 훈련된 모델의 신뢰도 점수도 각 예측에 대해 출력됩니다.

```
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."
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 엔티티를 참조하세요.

- 자세한 API 내용은 명령 참조DetectEntities의 섹션을 참조하세요. AWS CLI

### detect-key-phrases

다음 코드 예시에서는 detect-key-phrases을 사용하는 방법을 보여 줍니다.

AWS CLI

입력 텍스트에서 핵심 문구 탐지

다음 detect-key-phrases 예제에서는 입력 텍스트를 분석하고 핵심 명사구를 식별합니다. 사전 훈련된 모델의 신뢰도 점수도 각 예측에 대해 출력됩니다.

```
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."
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 핵심 문구를 참조하세요.

- 자세한 API 내용은 명령 참조DetectKeyPhrases의 섹션을 참조하세요. AWS CLI

### `detect-pii-entities`

다음 코드 예시에서는 `detect-pii-entities`을 사용하는 방법을 보여 줍니다.

AWS CLI

입력 텍스트에서 pii 엔티티 탐지

다음 `detect-pii-entities` 예제에서는 입력 텍스트를 분석하고 개인 식별 정보()가 포함된 엔터티를 식별합니다PII. 사전 훈련된 모델의 신뢰도 점수도 각 예측에 대해 출력됩니다.

```
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."
```

출력:


```
{
    "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
        }
    ]
}
```

자세한 내용은 Amazon Comprehend 개발자 안내서의 개인 식별 정보(PII)를 참조하세요.

- 자세한 API 내용은 명령 참조DetectPiiEntities의 섹션을 참조하세요. AWS CLI

### `detect-sentiment`

다음 코드 예시에서는 `detect-sentiment`을 사용하는 방법을 보여 줍니다.

AWS CLI

입력 텍스트의 감정 탐지

다음 `detect-sentiment` 예제는 입력 텍스트를 분석하고 일반적인 감정(`POSITIVE`, `NEUTRAL`, `MIXED` 또는 `NEGATIVE`)에 대한 추론을 반환합니다.

```
aws comprehend detect-sentiment \
    --language-code en \
    --text "It is a beautiful day in Seattle"
```

출력:

```
{
    "Sentiment": "POSITIVE",
    "SentimentScore": {
        "Positive": 0.9976957440376282,
        "Negative": 9.653854067437351e-05,
```

```
        "Neutral": 0.002169104292988777,
        "Mixed": 3.857641786453314e-05
    }
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 감정을 참조하세요

- 자세한 API 내용은 명령 참조DetectSentiment의 섹션을 참조하세요. AWS CLI

### `detect-syntax`

다음 코드 예시에서는 `detect-syntax`을 사용하는 방법을 보여 줍니다.

AWS CLI

입력 텍스트에서 품사 탐지

다음 `detect-syntax` 예제에서는 입력 텍스트의 구문을 분석하고 품사의 여러 부분을 반환합니다. 사전 훈련된 모델의 신뢰도 점수도 각 예측에 대해 출력됩니다.

```
aws comprehend detect-syntax \
    --language-code en \
    --text "It is a beautiful day in Seattle."
```

출력:

```
{
    "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
            }
        }
    ]
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 구문 분석을 참조하세요.

- 자세한 API 내용은 명령 참조DetectSyntax의 섹션을 참조하세요. AWS CLI

### `detect-targeted-sentiment`

다음 코드 예시에서는 `detect-targeted-sentiment`을 사용하는 방법을 보여 줍니다.

AWS CLI

입력 텍스트에서 명명된 엔터티의 대상 감정을 감지하려면

다음 `detect-targeted-sentiment` 예제에서는 입력 텍스트를 분석하고 각 엔터티와 연결된 대상 감정 외에도 명명된 엔터티를 반환합니다. 각 예측에 대해 사전 훈련된 모델 신뢰도 점수도 출력됩니다.

```
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"
```

출력:

```
{
    "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
                }
            ]
        }
    ]
}
```

자세한 내용은 Amazon Comprehend 개발자 안내서의 대상 감정을 참조하세요.

- 자세한 API 내용은 명령 참조DetectTargetedSentiment의 섹션을 참조하세요. AWS CLI

### `import-model`

다음 코드 예시에서는 `import-model`을 사용하는 방법을 보여 줍니다.

AWS CLI

모델을 가져오려면

다음 `import-model` 예제에서는 다른 AWS 계정에서 모델을 가져옵니다. 계정의 문서 분류기 모델444455556666에는 계정이 모델을 가져올 111122223333 수 있도록 허용하는 리소스 기반 정책이 있습니다.

```
aws comprehend import-model \
    --source-model-arn arn:aws:comprehend:us-west-2:444455556666:document-
classifier/example-classifier
```

출력:

```
{
    "ModelArn": "arn:aws:comprehend:us-west-2:111122223333:document-classifier/
example-classifier"
}
```

자세한 내용은 Amazon Comprehend 개발자 안내서의 AWS 계정 간에 사용자 지정 모델 복사를 참조하세요.

- 자세한 API 내용은 명령 참조ImportModel의 섹션을 참조하세요. AWS CLI

### list-datasets

다음 코드 예시에서는 list-datasets을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 플라이휠 데이터 세트를 나열하려면

다음 list-datasets 예제에서는 플라이휠과 연결된 모든 데이터 세트를 나열합니다.

```
aws comprehend list-datasets \
    --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/flywheel-
entity
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Comprehend 개발자 안내서의 플라이휠 개요를 참조하세요.

- 자세한 API 내용은 명령 참조ListDatasets의 섹션을 참조하세요. AWS CLI

### `list-document-classification-jobs`

다음 코드 예시에서는 `list-document-classification-jobs`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 문서 분류 작업 나열

다음 `list-document-classification-jobs` 예제에는 모든 문서 분류 작업이 나열되어 있습니다.

```
aws comprehend list-document-classification-jobs
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 사용자 지정 분류를 참조하세요.

- 자세한 API 내용은 명령 참조ListDocumentClassificationJobs의 섹션을 참조하세요. AWS CLI

### `list-document-classifier-summaries`

다음 코드 예시에서는 `list-document-classifier-summaries`을 사용하는 방법을 보여 줍니다.

AWS CLI

생성된 모든 문서 분류자의 요약을 나열하려면

다음 `list-document-classifier-summaries` 예제에서는 생성된 모든 문서 분류기 요약을 나열합니다.

```
aws comprehend list-document-classifier-summaries
```

출력:

```
{
```

```
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 사용자 지정 모델 생성 및 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ListDocumentClassifierSummaries의 섹션을 참조하세요. AWS CLI

### `list-document-classifiers`

다음 코드 예시에서는 `list-document-classifiers`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 문서 분류 작업 나열

다음 `list-document-classifiers` 예제에는 학습된 문서 분류자 모델과 학습 중인 문서 분류자 모델이 모두 나열되어 있습니다.

```
aws comprehend list-document-classifiers
```

출력:

```
{
    "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"
```

```
        }
    ]
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 사용자 지정 모델 생성 및 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ListDocumentClassifiers의 섹션을 참조하세요. AWS CLI

### `list-dominant-language-detection-jobs`

다음 코드 예시에서는 `list-dominant-language-detection-jobs`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 주요 언어 감지 작업을 나열하려면

다음 `list-dominant-language-detection-jobs` 예제에서는 진행 중인 작업과 완료된 비동기 우성 언어 감지 작업을 모두 나열합니다.

```
aws comprehend list-dominant-language-detection-jobs
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 Amazon Comprehend 통찰력 비동기 분석을 참조하십시오.

- 자세한 API 내용은 명령 참조ListDominantLanguageDetectionJobs의 섹션을 참조하세요. AWS CLI

### `list-endpoints`

다음 코드 예시에서는 `list-endpoints`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 엔드포인트를 나열하려면

다음 `list-endpoints` 예제에서는 모든 활성 모델별 엔드포인트를 나열합니다.

```
aws comprehend list-endpoints
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 Amazon Comprehend 엔드포인트 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ListEndpoints의 섹션을 참조하세요. AWS CLI

### list-entities-detection-jobs

다음 코드 예시에서는 list-entities-detection-jobs을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 개체 감지 작업을 나열하려면

다음 list-entities-detection-jobs 예제에서는 모든 비동기 엔터티 감지 작업을 나열합니다.

```
aws comprehend list-entities-detection-jobs
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 엔티티를 참조하세요.

- 자세한 API 내용은 명령 참조ListEntitiesDetectionJobs의 섹션을 참조하세요. AWS CLI

### `list-entity-recognizer-summaries`

다음 코드 예시에서는 `list-entity-recognizer-summaries`을 사용하는 방법을 보여 줍니다.

AWS CLI

생성된 모든 개체 인식기에 대한 요약을 나열하려면

다음 `list-entity-recognizer-summaries` 예제에서는 모든 엔터티 인식기 요약을 나열합니다.

```
aws comprehend list-entity-recognizer-summaries
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Comprehend 개발자 안내서의 사용자 지정 엔터티 인식을 참조하세요.

- 자세한 API 내용은 명령 참조ListEntityRecognizerSummaries의 섹션을 참조하세요. AWS CLI

### list-entity-recognizers

다음 코드 예시에서는 list-entity-recognizers을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 사용자 지정 엔터티 인식기 목록

다음 list-entity-recognizers 예제에서는 생성된 모든 사용자 지정 엔터티 인식기를 나열합니다.

```
aws comprehend list-entity-recognizers
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Comprehend 개발자 안내서의 사용자 지정 엔터티 인식을 참조하세요.

- 자세한 API 내용은 명령 참조ListEntityRecognizers의 섹션을 참조하세요. AWS CLI

### `list-events-detection-jobs`

다음 코드 예시에서는 `list-events-detection-jobs`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 이벤트 감지 작업을 나열하려면

다음 `list-events-detection-jobs` 예제에서는 모든 비동기 이벤트 감지 작업을 나열합니다.

```
aws comprehend list-events-detection-jobs
```

출력:

```
{
    "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"
            ]
        }
    ]
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 Amazon Comprehend 통찰력 비동기 분석을 참조하십시오.

- 자세한 API 내용은 명령 참조ListEventsDetectionJobs의 섹션을 참조하세요. AWS CLI

### `list-flywheel-iteration-history`

다음 코드 예시에서는 `list-flywheel-iteration-history`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 플라이휠 반복 기록을 나열하려면

다음 `list-flywheel-iteration-history` 예제에서는 플라이휠의 모든 반복을 나열합니다.

```
aws comprehend list-flywheel-iteration-history
    --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/example-
flywheel
```

출력:

```
{
    "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/"
        }
    ]
}
```

자세한 내용은 Amazon Comprehend 개발자 안내서의 Flywheel 개요를 참조하세요.

- 자세한 API 내용은 명령 참조ListFlywheelIterationHistory의 섹션을 참조하세요. AWS CLI

### `list-flywheels`

다음 코드 예시에서는 `list-flywheels`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 플라이휠을 나열하려면

다음 `list-flywheels` 예제에서는 생성된 모든 플라이휠을 나열합니다.

```
aws comprehend list-flywheels
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Comprehend 개발자 안내서의 Flywheel 개요를 참조하세요.

- 자세한 API 내용은 명령 참조ListFlywheels의 섹션을 참조하세요. AWS CLI

### `list-key-phrases-detection-jobs`

다음 코드 예시에서는 `list-key-phrases-detection-jobs`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 키 구문 감지 작업을 나열하려면

다음 `list-key-phrases-detection-jobs` 예제에서는 진행 중 및 완료된 비동기 키 구문 감지 작업을 모두 나열합니다.

```
aws comprehend list-key-phrases-detection-jobs
```

출력:

```
{
    "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"
```

```
        }
    ]
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 Amazon Comprehend 통찰력 비동기 분석을 참조하십시오.

- 자세한 API 내용은 명령 참조ListKeyPhrasesDetectionJobs의 섹션을 참조하세요. AWS CLI

### `list-pii-entities-detection-jobs`

다음 코드 예시에서는 `list-pii-entities-detection-jobs`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 pii 엔터티 감지 작업을 나열하려면

다음 `list-pii-entities-detection-jobs` 예제에서는 진행 중인 작업과 완료된 비동기 pii 감지 작업을 모두 나열합니다.

```
aws comprehend list-pii-entities-detection-jobs
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 Amazon Comprehend 통찰력 비동기 분석을 참조하십시오.

- 자세한 API 내용은 명령 참조ListPiiEntitiesDetectionJobs의 섹션을 참조하세요. AWS CLI

### `list-sentiment-detection-jobs`

다음 코드 예시에서는 `list-sentiment-detection-jobs`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 감정 감지 작업을 나열하려면

다음 `list-sentiment-detection-jobs` 예제에서는 진행 중인 작업과 완료된 비동기 감정 감지 작업을 모두 나열합니다.

```
aws comprehend list-sentiment-detection-jobs
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 Amazon Comprehend 통찰력 비동기 분석을 참조하십시오.

- 자세한 API 내용은 명령 참조ListSentimentDetectionJobs의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 대한 태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 Amazon Comprehend 리소스의 태그를 나열합니다.

```
aws comprehend list-tags-for-resource \
    --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/
example-classifier/version/1
```

출력:

```
{
    "ResourceArn": "arn:aws:comprehend:us-west-2:111122223333:document-classifier/
example-classifier/version/1",
    "Tags": [
        {
            "Key": "Department",
            "Value": "Finance"
        },
        {
            "Key": "location",
            "Value": "Seattle"
        }
    ]
```

```
}
```

자세한 내용은 Amazon Comprehend 개발자 안내서의 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### `list-targeted-sentiment-detection-jobs`

다음 코드 예시에서는 `list-targeted-sentiment-detection-jobs`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 대상 감정 감지 작업을 나열하려면

다음 `list-targeted-sentiment-detection-jobs` 예제에서는 진행 중인 작업과 완료된 비동기 표적 감정 감지 작업을 모두 나열합니다.

```
aws comprehend list-targeted-sentiment-detection-jobs
```

출력:

```
{
    "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-IOrole"
```

```
        },
        {
            "JobId": "123456abcdeb0e11022f22a1EXAMPLE2",
            "JobArn": "arn:aws:comprehend:us-west-2:111122223333:targeted-sentiment-
detection-job/123456abcdeb0e11022f22a1EXAMPLE2",
            "JobName": "example-targeted-sentiment-detection-job-2",
            "JobStatus": "COMPLETED",
            "SubmitTime": "2023-06-09T23:16:15.956000+00:00",
            "EndTime": "2023-06-09T23:26:00.168000+00:00",
            "InputDataConfig": {
                "S3Uri": "s3://DOC-EXAMPLE-BUCKET/MovieData2",
                "InputFormat": "ONE_DOC_PER_LINE"
            },
            "OutputDataConfig": {
                "S3Uri": "s3://DOC-EXAMPLE-DESTINATION-BUCKET/
testfolder/111122223333-TS-123456abcdeb0e11022f22a1EXAMPLE2/output/output.tar.gz"
            },
            "LanguageCode": "en",
            "DataAccessRoleArn": "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role"
        }
    ]
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 Amazon Comprehend 통찰력 비동기 분석을 참조하십시오.

- 자세한 API 내용은 명령 참조ListTargetedSentimentDetectionJobs의 섹션을 참조하세요. AWS CLI

### `list-topics-detection-jobs`

다음 코드 예시에서는 `list-topics-detection-jobs`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 주제 탐지 작업 나열

다음 `list-topics-detection-jobs` 예제는 진행 중인 모든 비동기 주제 탐지 작업과 완료된 비동기 주제 탐지 작업을 나열합니다.

```
aws comprehend list-topics-detection-jobs
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 Amazon Comprehend 통찰력 비동기 분석을 참조하십시오.

- 자세한 API 내용은 명령 참조ListTopicsDetectionJobs의 섹션을 참조하세요. AWS CLI

### put-resource-policy

다음 코드 예시에서는 put-resource-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 기반 정책을 연결하려면

다음 put-resource-policy 예제에서는 다른 AWS 계정으로 가져올 수 있도록 리소스 기반 정책을 모델에 연결합니다. 정책은 계정의 모델에 연결111122223333되며 계정을 통해 모델을 444455556666 가져올 수 있습니다.

```
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"]}}]}'
```

Ouput:

```
{
    "PolicyRevisionId": "aaa111d069d07afaa2aa3106aEXAMPLE"
}
```

자세한 내용은 Amazon Comprehend 개발자 안내서의 AWS 계정 간에 사용자 지정 모델 복사를 참조하세요.

- 자세한 API 내용은 명령 참조PutResourcePolicy의 섹션을 참조하세요. AWS CLI

### start-document-classification-job

다음 코드 예시에서는 start-document-classification-job을 사용하는 방법을 보여 줍니다.

AWS CLI

문서 분류 작업 나열

다음 start-document-classification-job 예제에서는 --input-data-config 태그로 지정된 주소의 모든 파일에서 사용자 지정 모델을 사용하여 문서 분류 작업을 시작합니다. 이 예제에서 입력 S3 버킷에는 SampleSMStext1.txt, SampleSMStext2.txt 및 SampleSMStext3.txt가 포함되어 있습니다. 이 모델은 이전에 스팸 및 스팸이 아닌 메시지 또는 “ham” SMS 메시지의 문서 분류에 대해 훈련되었습니다. 작업이 완료되면 --output-data-config 태그에 지정된 위치에 output.tar.gz가 배치됩니다. output.tar.gz에는 각 문서의 분류가 나열되는 predictions.jsonl이 들어 있습니다. Json 출력은 파일당 한 줄로 인쇄되지만 여기서는 가독성을 위해 형식이 지정됩니다.

```
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
```

SampleSMStext1.txt의 콘텐츠:

```
"CONGRATULATIONS! TXT 2155550100 to win $5000"
```

SampleSMStext2.txt의 콘텐츠:

```
"Hi, when do you want me to pick you up from practice?"
```

SampleSMStext3.txt의 콘텐츠:

```
"Plz send bank account # to 2155550100 to claim prize!!"
```

출력:

```
{
    "JobId": "e758dd56b824aa717ceab551fEXAMPLE",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:document-classification-
job/e758dd56b824aa717ceab551fEXAMPLE",
    "JobStatus": "SUBMITTED"
}
```

predictions.jsonl의 콘텐츠:

```
{"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}]}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 사용자 지정 분류를 참조하세요.

- 자세한 API 내용은 명령 참조StartDocumentClassificationJob의 섹션을 참조하세요. AWS CLI

### start-dominant-language-detection-job

다음 코드 예시에서는 start-dominant-language-detection-job을 사용하는 방법을 보여 줍니다.

AWS CLI

비동기 언어 감지 작업을 시작하려면

다음 `start-dominant-language-detection-job` 예제에서는 `--input-data-config` 태그에 지정된 주소에 있는 모든 파일에 대해 비동기 언어 감지 작업을 시작합니다. 이 예제의 S3 버킷에는 이 포함되어 있습니다`Sampletext1.txt`. 작업이 완료되면 폴더 가 `--output-data-config` 태그에 지정된 위치에 배치`output`됩니다. 폴더에는 각 텍스트 파일의 지배적 언어와 각 예측에 대해 사전 훈련된 모델의 신뢰도 점수가 `output.txt` 포함된 가 포함되어 있습니다.

```
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
```

Sampletext1.txt의 내용:

```
"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."
```

출력:

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:dominant-language-
detection-job/123456abcdeb0e11022f22a11EXAMPLE",
    "JobStatus": "SUBMITTED"
}
```

`output.txt`의 콘텐츠:

```
{"File": "Sampletext1.txt", "Languages": [{"LanguageCode": "en", "Score":
 0.9913753867149353}], "Line": 0}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 Amazon Comprehend 통찰력 비동기 분석을 참조하십시오.

- 자세한 API 내용은 명령 참조StartDominantLanguageDetectionJob의 섹션을 참조하세요. AWS CLI

### start-entities-detection-job

다음 코드 예시에서는 start-entities-detection-job을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 사전 훈련된 모델을 사용하여 표준 개체 감지 작업을 시작하려면

다음 start-entities-detection-job 예제에서는 --input-data-config 태그에 지정된 주소에 있는 모든 파일에 대해 비동기 엔터티 감지 작업을 시작합니다. 이 예제의 S3 버킷에는 Sampletext1.txt, 및 Sampletext2.txt가 포함되어 있습니다Sampletext3.txt. 작업이 완료되면 폴더 가 --output-data-config 태그에 지정된 위치에 배치output됩니다. 폴더에는 각 텍스트 파일 내에서 감지된 모든 명명된 엔터티와 각 예측에 대해 사전 훈련된 모델의 신뢰도 점수를 나열output.txt하는 항목이 포함되어 있습니다. Json 출력은 입력 파일당 한 줄에 인쇄되지만 가독성을 위해 여기에 포맷되어 있습니다.

```
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
```

Sampletext1.txt의 콘텐츠:

```
"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."
```

Sampletext2.txt의 콘텐츠:

```
"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. "
```

Sampletext3.txt의 콘텐츠:

```
"Jane, please submit any customer feedback from this weekend to AnySpa, 123 Main St,
 Anywhere and send comments to Alice at AnySpa@example.com."
```

출력:

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:entities-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
    "JobStatus": "SUBMITTED"
}
```

가독성을 위해 줄 바꿈이 output.txt 있는 의 내용:

```
{
"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
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 Amazon Comprehend 통찰력 비동기 분석을 참조하십시오.

예제 2: 사용자 지정 개체 감지 작업을 시작하려면

다음 start-entities-detection-job 예제에서는 --input-data-config 태그에 지정된 주소에 있는 모든 파일에 대해 비동기 사용자 지정 엔터티 감지 작업을 시작합니다. 이 예제에서는 이 예제의 S3 버킷에 SampleFeedback1.txt, 및 SampleFeedback2.txt가 포함되어 있습니다SampleFeedback3.txt. 엔터티 인식기 모델은 디바이스 이름을 인식하기 위해 고객 지원 피드백에 대해 훈련되었습니다. 작업이 완료되면 폴더 가 --output-data-config 태그에 지정된 위치에 배치output됩니다. 폴더에는 각 텍스트 파일 내에서 감지된 모든 명명된 엔터티와 각 예측에 대해 사전 훈련된 모델의 신뢰도 점수를 나열output.txt하는 가 포함되어 있습니다. Json 출력은 파일당 한 줄로 인쇄되지만 여기서는 가독성을 위해 형식이 지정됩니다.

```
aws comprehend start-entities-detection-job \
    --job-name customentitiestest \
    --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-IOrole"
```

SampleFeedback1.txt의 콘텐츠:

```
"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!"
```

SampleFeedback2.txt의 콘텐츠:

```
"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!"
```

SampleFeedback3.txt의 콘텐츠:

```
"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!"
```

출력:

```
{
    "JobId": "019ea9edac758806850fa8a79ff83021",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:entities-detection-
job/019ea9edac758806850fa8a79ff83021",
    "JobStatus": "SUBMITTED"
}
```

가독성을 위해 줄 바꿈이 output.txt 있는 의 내용:

```
{
"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
}
```

자세한 내용은 Amazon Comprehend 개발자 안내서의 사용자 지정 엔터티 인식을 참조하세요.

- 자세한 API 내용은 명령 참조StartEntitiesDetectionJob의 섹션을 참조하세요. AWS CLI

### start-events-detection-job

다음 코드 예시에서는 start-events-detection-job을 사용하는 방법을 보여 줍니다.

AWS CLI

비동기 이벤트 감지 작업을 시작하려면

다음 start-events-detection-job 예제에서는 --input-data-config 태그에 지정된 주소에 있는 모든 파일에 대해 비동기 이벤트 감지 작업을 시작합니다. 가능한 대상 이벤트 유형에는 BANKRUPCTY, , EMPLOYMENT, CORPORATE_ACQUISITION, INVESTMENT_GENERAL, CORPORATE_MERGER, IPORIGHTS_ISSUE, SECONDARY_OFFERING, SHELF_OFFERING, TENDER_OFFERING, 및 가 포함됩니다STOCK_SPLIT. 이 예제의 S3 버킷에는 SampleText1.txt, 및 SampleText2.txt가 포함되어 있습니다SampleText3.txt. 작업이 완료되면 폴더 output가 --output-data-config 태그에 지정된 위치에 배치됩니다. 폴더에는 SampleText1.txt.out, 및 SampleText2.txt.out가 포함되어 있습니다SampleText3.txt.out. JSON 출력은 파일당 한 줄에 인쇄되지만, 가독성을 위해 여기에 형식이 지정됩니다.

```
aws comprehend start-events-detection-job \
    --job-name events-detection-1 \
    --input-data-config "S3Uri=s3://DOC-EXAMPLE-BUCKET/EventsData" \
    --output-data-config "S3Uri=s3://DOC-EXAMPLE-DESTINATION-BUCKET/testfolder/" \
    --data-access-role-arn arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-servicerole \
    --language-code en \
    --target-event-
types "BANKRUPTCY" "EMPLOYMENT" "CORPORATE_ACQUISITION" "CORPORATE_MERGER" "INVESTMENT_GENER
```

SampleText1.txt의 콘텐츠:

```
"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."
```

SampleText2.txt의 콘텐츠:

```
"In 2021, AnyCompany officially purchased AnyBusiness for 100 billion dollars,
 surprising and exciting the shareholders."
```

SampleText3.txt의 콘텐츠:

```
"In 2022, AnyCompany stock crashed 50. Eventually later that year they filed for
 bankruptcy."
```

출력:

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:events-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
    "JobStatus": "SUBMITTED"
}
```

가독성을 위한 줄 들여쓰기가 SampleText1.txt.out 있는 의 내용:

```
{
    "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": "INVESTEE",
            "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
}
```

SampleText2.txt.out의 콘텐츠:

```
{
    "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": "INVESTEE",
            "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
}
```

SampleText3.txt.out의 콘텐츠:

```
{
    "Entities": [
        {
```

```
        "Mentions": [
            {
            "BeginOffset": 9,
            "EndOffset": 19,
            "Score": 0.999774,
            "Text": "AnyCompany",
            "Type": "ORGANIZATION",
            "GroupScore": 1
            },
            {
            "BeginOffset": 66,
            "EndOffset": 70,
            "Score": 0.995717,
            "Text": "they",
            "Type": "ORGANIZATION",
            "GroupScore": 0.997626
            }
        ]
        },
        {
        "Mentions": [
            {
            "BeginOffset": 50,
            "EndOffset": 65,
            "Score": 0.999656,
            "Text": "later that year",
            "Type": "DATE",
            "GroupScore": 1
            }
        ]
        }
    ],
    "Events": [
        {
        "Type": "BANKRUPTCY",
        "Arguments": [
            {
            "EntityIndex": 1,
            "Role": "DATE",
            "Score": 0.999656
            },
            {
            "EntityIndex": 0,
            "Role": "FILER",
```

```
            "Score": 0.995717
            }
        ],
        "Triggers": [
            {
            "BeginOffset": 81,
            "EndOffset": 91,
            "Score": 0.999936,
            "Text": "bankruptcy",
            "Type": "BANKRUPTCY",
            "GroupScore": 1
            }
        ]
        }
    ],
    "File": "SampleText3.txt",
    "Line": 0
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 Amazon Comprehend 통찰력 비동기 분석을 참조하십시오.

- 자세한 API 내용은 명령 참조StartEventsDetectionJob의 섹션을 참조하세요. AWS CLI

### `start-flywheel-iteration`

다음 코드 예시에서는 `start-flywheel-iteration`을 사용하는 방법을 보여 줍니다.

AWS CLI

플라이휠 반복을 시작하려면

다음 `start-flywheel-iteration` 예제에서는 플라이휠 반복을 시작합니다. 이 작업은 플라이휠의 새 데이터 세트를 사용하여 새 모델 버전을 훈련합니다.

```
aws comprehend start-flywheel-iteration \
    --flywheel-arn arn:aws:comprehend:us-west-2:111122223333:flywheel/example-flywheel
```

출력:

```
{
```

```
    "FlywheelArn": "arn:aws:comprehend:us-west-2:111122223333:flywheel/example-
flywheel",
    "FlywheelIterationId": "12345123TEXAMPLE"
}
```

자세한 내용은 Amazon Comprehend 개발자 안내서의 플라이휠 개요를 참조하세요.

- 자세한 API 내용은 명령 참조StartFlywheelIteration의 섹션을 참조하세요. AWS CLI

### start-key-phrases-detection-job

다음 코드 예시에서는 start-key-phrases-detection-job을 사용하는 방법을 보여 줍니다.

AWS CLI

키 구문 감지 작업을 시작하려면

다음 start-key-phrases-detection-job 예제에서는 --input-data-config 태그에 지정된 주소에 있는 모든 파일에 대해 비동기 키 구문 감지 작업을 시작합니다. 이 예제의 S3 버킷에는 Sampletext1.txt, 및 Sampletext2.txt가 포함되어 있습니다Sampletext3.txt. 작업이 완료되면 폴더 가 --output-data-config 태그에 지정된 위치에 배치output됩니다. 폴더에는 각 텍스트 파일 내에서 감지output.txt된 모든 키 구문과 각 예측에 대해 사전 훈련된 모델의 신뢰도 점수가 포함된 파일이 포함되어 있습니다. Json 출력은 파일당 한 줄로 인쇄되지만 여기서는 가독성을 위해 형식이 지정됩니다.

```
aws comprehend start-key-phrases-detection-job \
    --job-name keyphrasesanalysistest1 \
    --language-code en \
    --input-data-config "S3Uri=s3://DOC-EXAMPLE-BUCKET/" \
    --output-data-config "S3Uri=s3://DOC-EXAMPLE-DESTINATION-BUCKET/testfolder/" \
    --data-access-role-arn "arn:aws:iam::111122223333:role/service-role/
AmazonComprehendServiceRole-example-role" \
    --language-code en
```

Sampletext1.txt의 콘텐츠:

```
"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."
```

Sampletext2.txt의 콘텐츠:

```
"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. "
```

Sampletext3.txt의 콘텐츠:

```
"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."
```

출력:

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:key-phrases-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
    "JobStatus": "SUBMITTED"
}
```

가독성을 위해 줄 바꿈이 output.txt 있는 의 내용:

```
{
    "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
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 Amazon Comprehend 통찰력 비동기 분석을 참조하십시오.

- 자세한 API 내용은 명령 참조StartKeyPhrasesDetectionJob의 섹션을 참조하세요. AWS CLI

### start-pii-entities-detection-job

다음 코드 예시에서는 start-pii-entities-detection-job을 사용하는 방법을 보여 줍니다.

AWS CLI

비동기 PII 감지 작업을 시작하려면

다음 start-pii-entities-detection-job 예제에서는 --input-data-config 태그에 지정된 주소에 있는 모든 파일에 대해 비동기 개인 식별 정보(PII) 엔터티 감지 작업을 시작합니다. 이 예제의 S3 버킷에는 Sampletext1.txt, 및 Sampletext2.txt가 포함되어 있습니다Sampletext3.txt. 작업이 완료되면 폴더 가 --output-data-config 태그에 지정된 위치에 배치output됩니다. 폴더에는 각 텍스트 파일 내에 이름이 지정된 엔터티를 나열SampleText3.txt.out하는 SampleText1.txt.outSampleText2.txt.out, 및 가 포함되어 있습니다. Json 출력은 파일당 한 줄로 인쇄되지만 여기서는 가독성을 위해 형식이 지정됩니다.

```
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
```

Sampletext1.txt의 콘텐츠:

```
"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."
```

Sampletext2.txt의 콘텐츠:

```
"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. "
```

Sampletext3.txt의 콘텐츠:

```
"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."
```

출력:

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:pii-entities-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
    "JobStatus": "SUBMITTED"
}
```

가독성을 위한 줄 들여쓰기가 SampleText1.txt.out 있는 의 내용:

```
{
    "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
}
```

가독성을 위해 줄 바꿈이 SampleText2.txt.out 있는 의 내용:

```
{
    "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
}
```

가독성을 위한 줄 들여쓰기가 SampleText3.txt.out 있는 의 내용:

```
{
    "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
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 Amazon Comprehend 통찰력 비동기 분석을 참조하십시오.

- 자세한 API 내용은 명령 참조StartPiiEntitiesDetectionJob의 섹션을 참조하세요. AWS CLI

### `start-sentiment-detection-job`

다음 코드 예시에서는 `start-sentiment-detection-job`을 사용하는 방법을 보여 줍니다.

AWS CLI

비동기 감정 분석 작업을 시작하려면

다음 `start-sentiment-detection-job` 예제에서는 `--input-data-config` 태그에 지정된 주소에 있는 모든 파일에 대해 비동기 감정 분석 감지 작업을 시작합니다. 이 예제의 S3 버

킷 폴더에는 SampleMovieReview1.txt, 및 SampleMovieReview2.txt가 포함되어 있습니다SampleMovieReview3.txt. 작업이 완료되면 폴더 output가 --output-data-config 태그에 지정된 위치에 배치됩니다. 폴더에는 각 텍스트 파일의 일반적인 감성과 각 예측에 대해 사전 훈련된 모델의 신뢰도 점수가 포함된 파일 output.txt가 포함되어 있습니다. Json 출력은 파일당 한 줄로 인쇄되지만 여기서는 가독성을 위해 형식이 지정됩니다.

```
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
```

SampleMovieReview1.txt의 콘텐츠:

```
"The film, AnyMovie2, is fairly predictable and just okay."
```

SampleMovieReview2.txt의 콘텐츠:

```
"AnyMovie2 is the essential sci-fi film that I grew up watching when I was a kid. I
 highly recommend this movie."
```

SampleMovieReview3.txt의 콘텐츠:

```
"Don't get fooled by the 'awards' for AnyMovie2. All parts of the film were poorly
 stolen from other modern directors."
```

출력:

```
{
    "JobId": "0b5001e25f62ebb40631a9a1a7fde7b3",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:sentiment-detection-
job/0b5001e25f62ebb40631a9a1a7fde7b3",
    "JobStatus": "SUBMITTED"
}
```

가독성을 위해 들여쓰기 줄이 output.txt 있는 의 내용:

```
{
```

```
    "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
        }
    }
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 Amazon Comprehend 통찰력 비동기 분석을 참조하십시오.

- 자세한 API 내용은 명령 참조StartSentimentDetectionJob의 섹션을 참조하세요. AWS CLI

### `start-targeted-sentiment-detection-job`

다음 코드 예시에서는 `start-targeted-sentiment-detection-job`을 사용하는 방법을 보여 줍니다.

AWS CLI

비동기 표적 감정 분석 작업을 시작하려면

다음 start-targeted-sentiment-detection-job 예제에서는 --input-data-config 태그에 지정된 주소에 있는 모든 파일에 대해 비동기 표적 감정 분석 감지 작업을 시작합니다. 이 예제의 S3 버킷 폴더에는 SampleMovieReview1.txt, 및 SampleMovieReview2.txt가 포함되어 있습니다SampleMovieReview3.txt. 작업이 완료되면 output.tar.gz는 --output-data-config 태그에 의해 지정된 위치에 배치됩니다. output.tar.gz에는 파일 SampleMovieReview1.txt.out, 및 SampleMovieReview2.txt.out가 포함되며SampleMovieReview3.txt.out, 각 파일에는 단일 입력 텍스트 파일에 대한 명명된 엔터티 및 관련 감정이 모두 포함됩니다.

```
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
```

SampleMovieReview1.txt의 콘텐츠:

```
"The film, AnyMovie, is fairly predictable and just okay."
```

SampleMovieReview2.txt의 콘텐츠:

```
"AnyMovie is the essential sci-fi film that I grew up watching when I was a kid. I
 highly recommend this movie."
```

SampleMovieReview3.txt의 콘텐츠:

```
"Don't get fooled by the 'awards' for AnyMovie. All parts of the film were poorly
 stolen from other modern directors."
```

출력:

```
{
    "JobId": "0b5001e25f62ebb40631a9a1a7fde7b3",
```

```
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:targeted-sentiment-
detection-job/0b5001e25f62ebb40631a9a1a7fde7b3",
    "JobStatus": "SUBMITTED"
}
```

가독성을 위한 줄 들여쓰기가 SampleMovieReview1.txt.out 있는 의 내용:

```
{
    "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
}
```

가독성을 위한 SampleMovieReview2.txt.out 라인 들여쓰기의 내용:

```
{
    "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
}
```

가독성을 위해 줄 바꿈이 SampleMovieReview3.txt.out 있는 의 내용:

```
{
    "Entities": [
```

```
        {
        "DescriptiveMentionIndex": [
            1
        ],
        "Mentions": [
            {
            "BeginOffset": 64,
            "EndOffset": 68,
            "Score": 0.992953,
            "GroupScore": 0.999814,
            "Text": "film",
            "Type": "MOVIE",
            "MentionSentiment": {
                "Sentiment": "NEUTRAL",
                "SentimentScore": {
                "Mixed": 0.000004,
                "Negative": 0.010425,
                "Neutral": 0.989543,
                "Positive": 0.000027
                }
            }
            },
            {
            "BeginOffset": 37,
            "EndOffset": 45,
            "Score": 0.999782,
            "GroupScore": 1,
            "Text": "AnyMovie",
            "Type": "ORGANIZATION",
            "MentionSentiment": {
                "Sentiment": "POSITIVE",
                "SentimentScore": {
                "Mixed": 0.000095,
                "Negative": 0.039847,
                "Neutral": 0.000673,
                "Positive": 0.959384
                }
            }
            }
        ]
        },
        {
        "DescriptiveMentionIndex": [
            0
```

```
        ],
        "Mentions": [
            {
            "BeginOffset": 47,
            "EndOffset": 50,
            "Score": 0.999991,
            "GroupScore": 1,
            "Text": "All",
            "Type": "QUANTITY",
            "MentionSentiment": {
                "Sentiment": "NEUTRAL",
                "SentimentScore": {
                "Mixed": 0.000001,
                "Negative": 0.000001,
                "Neutral": 0.999998,
                "Positive": 0
                }
            }
            }
        ]
        },
        {
        "DescriptiveMentionIndex": [
            0
        ],
        "Mentions": [
            {
            "BeginOffset": 106,
            "EndOffset": 115,
            "Score": 0.542083,
            "GroupScore": 1,
            "Text": "directors",
            "Type": "PERSON",
            "MentionSentiment": {
                "Sentiment": "NEUTRAL",
                "SentimentScore": {
                "Mixed": 0,
                "Negative": 0,
                "Neutral": 1,
                "Positive": 0
                }
            }
            }
        ]
```

```
        }
    ],
    "File": "SampleMovieReview3.txt",
    "Line": 0
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 Amazon Comprehend 통찰력 비동기 분석을 참조하십시오.

- 자세한 API 내용은 명령 참조StartTargetedSentimentDetectionJob의 섹션을 참조하세요. AWS CLI

### start-topics-detection-job

다음 코드 예시에서는 start-topics-detection-job을 사용하는 방법을 보여 줍니다.

AWS CLI

주제 탐지 분석 작업 시작

다음 start-topics-detection-job 예제에서는 --input-data-config 태그로 지정된 주소에 있는 모든 파일에 대해 비동기 주제 탐지 작업을 시작합니다. 작업이 완료되면 --ouput-data-config 태그로 지정된 위치에 output 폴더가 배치됩니다. output에는 topic-terms.csv 및 doc-topics.csv 파일이 들어 있습니다. 첫 번째 출력 파일 topic-terms.csv는 컬렉션의 주제 목록입니다. 각 주제에 대해 목록에는 기본적으로 주제별 상위 용어가 가중치에 따라 포함됩니다. 두 번째 doc-topics.csv 파일에는 주제와 관련된 문서 및 해당 주제와 관련된 문서 비율이 나열되어 있습니다.

```
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
```

출력:

```
{
```

```
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE",
    "JobArn": "arn:aws:comprehend:us-west-2:111122223333:key-phrases-detection-
job/123456abcdeb0e11022f22a11EXAMPLE",
    "JobStatus": "SUBMITTED"
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 주제 모델링을 참조하세요.

- 자세한 API 내용은 명령 참조StartTopicsDetectionJob의 섹션을 참조하세요. AWS CLI

### stop-dominant-language-detection-job

다음 코드 예시에서는 stop-dominant-language-detection-job을 사용하는 방법을 보여 줍니다.

AWS CLI

비동기 우성 언어 감지 작업을 중지하려면

다음 stop-dominant-language-detection-job 예제에서는 진행 중인 비동기식 우성 언어 감지 작업을 중지합니다. 현재 작업 상태가 IN_PROGRESS인 경우 작업이 종료로 표시되고 STOP_REQUESTED 상태가 됩니다. 작업을 중지하기 전에 작업이 완료되면 COMPLETED 상태가 됩니다.

```
aws comprehend stop-dominant-language-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

출력:

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE,
    "JobStatus": "STOP_REQUESTED"
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 Amazon Comprehend 통찰력 비동기 분석을 참조하십시오.

- 자세한 API 내용은 명령 참조StopDominantLanguageDetectionJob의 섹션을 참조하세요. AWS CLI

### stop-entities-detection-job

다음 코드 예시에서는 stop-entities-detection-job을 사용하는 방법을 보여 줍니다.

AWS CLI

비동기 엔터티 감지 작업을 중지하려면

다음 stop-entities-detection-job 예제에서는 진행 중인 비동기 엔터티 감지 작업을 중지합니다. 현재 작업 상태가 IN_PROGRESS인 경우 작업이 종료로 표시되고 STOP_REQUESTED 상태가 됩니다. 작업을 중지하기 전에 작업이 완료되면 COMPLETED 상태가 됩니다.

```
aws comprehend stop-entities-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

출력:

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE,
    "JobStatus": "STOP_REQUESTED"
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 Amazon Comprehend 통찰력 비동기 분석을 참조하십시오.

- 자세한 API 내용은 명령 참조StopEntitiesDetectionJob의 섹션을 참조하세요. AWS CLI

### stop-events-detection-job

다음 코드 예시에서는 stop-events-detection-job을 사용하는 방법을 보여 줍니다.

AWS CLI

비동기 이벤트 감지 작업을 중지하려면

다음 stop-events-detection-job 예제에서는 진행 중인 비동기 이벤트 감지 작업을 중지합니다. 현재 작업 상태가 IN_PROGRESS인 경우 작업이 종료로 표시되고 STOP_REQUESTED 상태가 됩니다. 작업을 중지하기 전에 작업이 완료되면 COMPLETED 상태가 됩니다.

```
aws comprehend stop-events-detection-job \
```

```
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

출력:

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE,
    "JobStatus": "STOP_REQUESTED"
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 Amazon Comprehend 통찰력 비동기 분석을 참조하십시오.

- 자세한 API 내용은 명령 참조StopEventsDetectionJob의 섹션을 참조하세요. AWS CLI

### stop-key-phrases-detection-job

다음 코드 예시에서는 stop-key-phrases-detection-job을 사용하는 방법을 보여 줍니다.

AWS CLI

비동기식 키 문구 감지 작업을 중지하려면

다음 stop-key-phrases-detection-job 예제에서는 진행 중인 비동기식 키 구문 감지 작업을 중지합니다. 현재 작업 상태가 IN_PROGRESS인 경우 작업이 종료로 표시되고 STOP_REQUESTED 상태가 됩니다. 작업을 중지하기 전에 작업이 완료되면 COMPLETED 상태가 됩니다.

```
aws comprehend stop-key-phrases-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

출력:

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE,
    "JobStatus": "STOP_REQUESTED"
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 Amazon Comprehend 통찰력 비동기 분석을 참조하십시오.

- 자세한 API 내용은 명령 참조StopKeyPhrasesDetectionJob의 섹션을 참조하세요. AWS CLI

### stop-pii-entities-detection-job

다음 코드 예시에서는 stop-pii-entities-detection-job을 사용하는 방법을 보여 줍니다.

AWS CLI

비동기 pii 엔터티 감지 작업을 중지하려면

다음 stop-pii-entities-detection-job 예제에서는 진행 중인 비동기 pii 엔터티 감지 작업을 중지합니다. 현재 작업 상태가 IN_PROGRESS인 경우 작업이 종료로 표시되고 STOP_REQUESTED 상태가 됩니다. 작업을 중지하기 전에 작업이 완료되면 COMPLETED 상태가 됩니다.

```
aws comprehend stop-pii-entities-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

출력:

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE,
    "JobStatus": "STOP_REQUESTED"
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 Amazon Comprehend 통찰력 비동기 분석을 참조하십시오.

- 자세한 API 내용은 명령 참조StopPiiEntitiesDetectionJob의 섹션을 참조하세요. AWS CLI

### stop-sentiment-detection-job

다음 코드 예시에서는 stop-sentiment-detection-job을 사용하는 방법을 보여 줍니다.

AWS CLI

비동기 감정 감지 작업을 중지하려면

다음 stop-sentiment-detection-job 예제에서는 진행 중인 비동기 감정 감지 작업을 중지합니다. 현재 작업 상태가 IN_PROGRESS인 경우 작업이 종료로 표시되고 STOP_REQUESTED 상태가 됩니다. 작업을 중지하기 전에 작업이 완료되면 COMPLETED 상태가 됩니다.

```
aws comprehend stop-sentiment-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

출력:

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE,
    "JobStatus": "STOP_REQUESTED"
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 Amazon Comprehend 통찰력 비동기 분석을 참조하십시오.

- 자세한 API 내용은 명령 참조StopSentimentDetectionJob의 섹션을 참조하세요. AWS CLI

### stop-targeted-sentiment-detection-job

다음 코드 예시에서는 stop-targeted-sentiment-detection-job을 사용하는 방법을 보여 줍니다.

AWS CLI

비동기 표적 감정 감지 작업을 중지하려면

다음 stop-targeted-sentiment-detection-job 예제에서는 진행 중인 비동기 표적 감정 감지 작업을 중지합니다. 현재 작업 상태가 IN_PROGRESS인 경우 작업이 종료로 표시되고 STOP_REQUESTED 상태가 됩니다. 작업을 중지하기 전에 작업이 완료되면 COMPLETED 상태가 됩니다.

```
aws comprehend stop-targeted-sentiment-detection-job \
    --job-id 123456abcdeb0e11022f22a11EXAMPLE
```

출력:

```
{
    "JobId": "123456abcdeb0e11022f22a11EXAMPLE,
    "JobStatus": "STOP_REQUESTED"
}
```

자세한 내용은 Amazon Comprehend 개발자 가이드의 Amazon Comprehend 통찰력 비동기 분석을 참조하십시오.

- 자세한 API 내용은 명령 참조StopTargetedSentimentDetectionJob의 섹션을 참조하세요. AWS CLI

### stop-training-document-classifier

다음 코드 예시에서는 stop-training-document-classifier을 사용하는 방법을 보여 줍니다.

AWS CLI

문서 분류기 모델의 훈련을 중지하려면

다음 stop-training-document-classifier 예제에서는 진행 중인 문서 분류기 모델의 훈련을 중지합니다.

```
aws comprehend stop-training-document-classifier
    --document-classifier-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/example-classifier
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Comprehend 개발자 가이드의 사용자 지정 모델 생성 및 관리를 참조하세요.

- 자세한 API 내용은 명령 참조StopTrainingDocumentClassifier의 섹션을 참조하세요. AWS CLI

### stop-training-entity-recognizer

다음 코드 예시에서는 stop-training-entity-recognizer을 사용하는 방법을 보여 줍니다.

AWS CLI

엔터티 인식기 모델의 훈련을 중지하려면

다음 stop-training-entity-recognizer 예제는 진행 중인 엔터티 인식기 모델의 훈련을 중지합니다.

```
aws comprehend stop-training-entity-recognizer
    --entity-recognizer-arn "arn:aws:comprehend:us-west-2:111122223333:entity-recognizer/examplerecognizer1"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Comprehend 개발자 가이드의 사용자 지정 모델 생성 및 관리를 참조하세요.

- 자세한 API 내용은 명령 참조StopTrainingEntityRecognizer의 섹션을 참조하세요. AWS CLI

### tag-resource

다음 코드 예시에서는 tag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 리소스에 태그 지정

다음 tag-resource 예제에서는 Amazon Comprehend 리소스에 단일 태그를 추가합니다.

```
aws comprehend tag-resource \
    --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/
example-classifier/version/1 \
    --tags Key=Location,Value=Seattle
```

이 명령에는 출력이 없습니다.

자세한 내용은 Amazon Comprehend 개발자 안내서의 리소스 태그 지정을 참조하세요.

예제 2: 리소스에 여러 태그를 추가하려면

다음 tag-resource 예제에서는 Amazon Comprehend 리소스에 여러 태그를 추가합니다.

```
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
```

이 명령에는 출력이 없습니다.

자세한 내용은 Amazon Comprehend 개발자 안내서의 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### untag-resource

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 리소스에서 단일 태그를 제거하려면

다음 untag-resource 예제에서는 Amazon Comprehend 리소스에서 단일 태그를 제거합니다.

```
aws comprehend untag-resource \
    --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/
example-classifier/version/1
    --tag-keys Location
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Comprehend 개발자 안내서의 리소스 태그 지정을 참조하세요.

예제 2: 리소스에서 여러 태그를 제거하려면

다음 `untag-resource` 예제에서는 Amazon Comprehend 리소스에서 여러 태그를 제거합니다.

```
aws comprehend untag-resource \
    --resource-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier/
example-classifier/version/1
    --tag-keys Location Department
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Comprehend 개발자 안내서의 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-endpoint

다음 코드 예시에서는 `update-endpoint`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 엔드포인트의 추론 단위 업데이트

다음 `update-endpoint` 예제에서는 엔드포인트에 대한 정보를 업데이트합니다. 이 예제에서는 추론 단위 수가 증가합니다.

```
aws comprehend update-endpoint \
    --endpoint-arn arn:aws:comprehend:us-west-2:111122223333:document-classifier-
endpoint/example-classifier-endpoint
    --desired-inference-units 2
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Comprehend 개발자 가이드의 Amazon Comprehend 엔드포인트 관리를 참조하세요.

예제 2: 엔드포인트의 액티 모델을 업데이트하려면

다음 update-endpoint 예제에서는 엔드포인트에 대한 정보를 업데이트합니다. 이 예제에서는 활성 모델이 변경됩니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Comprehend 개발자 가이드의 Amazon Comprehend 엔드포인트 관리를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateEndpoint의 섹션을 참조하세요. AWS CLI

### update-flywheel

다음 코드 예시에서는 update-flywheel을 사용하는 방법을 보여 줍니다.

AWS CLI

플라이휠 구성을 업데이트하려면

다음 update-flywheel 예제에서는 플라이휠 구성을 업데이트합니다. 이 예제에서는 플라이휠의 활성 모델이 업데이트됩니다.

```
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
```

출력:

```
{
```

```
    "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"
    }
}
```

자세한 내용은 Amazon Comprehend 개발자 안내서의 Flywheel 개요를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateFlywheel의 섹션을 참조하세요. AWS CLI

## 를 사용한 Amazon Comprehend Medical 예제 AWS CLI

다음 코드 예제에서는 Amazon Comprehend Medical과 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### `describe-entities-detection-v2-job`

다음 코드 예시에서는 `describe-entities-detection-v2-job`을 사용하는 방법을 보여 줍니다.

AWS CLI

엔터티 감지 작업을 설명하려면

다음 `describe-entities-detection-v2-job` 예제에서는 비동기 엔터티 감지 작업과 연결된 속성을 표시합니다.

```
aws comprehendmedical describe-entities-detection-v2-job \
    --job-id "ab9887877365fe70299089371c043b96"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Comprehend Medical 개발자 안내서의 배치APIs를 참조하세요.

- API 자세한 내용은 AWS CLI 명령 참조의 DescribeEntitiesDetectionV2Job을 참조하세요.

### `describe-icd10-cm-inference-job`

다음 코드 예시에서는 `describe-icd10-cm-inference-job`을 사용하는 방법을 보여 줍니다.

AWS CLI

ICD-10-CM 추론 작업을 설명하려면

다음 `describe-icd10-cm-inference-job` 예제에서는 지정된 작업 ID를 사용하여 요청된 추론 작업의 속성을 설명합니다.

```
aws comprehendmedical describe-icd10-cm-inference-job \
    --job-id "5780034166536cdb52ffa3295a1b00a7"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Comprehend Medical 개발자 안내서의 Ontology 연결 배치 분석을 참조하세요.

- API 자세한 내용은 AWS CLI 명령 참조의 DescribeIcd10CmInferenceJob을 참조하세요.

### describe-phi-detection-job

다음 코드 예시에서는 describe-phi-detection-job을 사용하는 방법을 보여 줍니다.

AWS CLI

PHI 감지 작업을 설명하려면

다음 describe-phi-detection-job 예제에서는 비동기식 보호된 상태 정보(PHI) 감지 작업과 연결된 속성을 표시합니다.

```
aws comprehendmedical describe-phi-detection-job \
    --job-id "4750034166536cdb52ffa3295a1b00a3"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Comprehend Medical 개발자 안내서의 배치APIs를 참조하세요.

- 자세한 API 내용은 명령 참조DescribePhiDetectionJob의 섹션을 참조하세요. AWS CLI

### describe-rx-norm-inference-job

다음 코드 예시에서는 describe-rx-norm-inference-job을 사용하는 방법을 보여 줍니다.

AWS CLI

RxNorm 추론 작업을 설명하려면

다음 describe-rx-norm-inference-job 예제에서는 지정된 job-id를 사용하여 요청된 추론 작업의 속성을 설명합니다.

```
aws comprehendmedical describe-rx-norm-inference-job \
    --job-id "eg8199877365fc70299089371c043b96"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Comprehend Medical 개발자 안내서의 온톨로지 연결 배치 분석을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeRxNormInferenceJob의 섹션을 참조하세요. AWS CLI

### describe-snomedct-inference-job

다음 코드 예시에서는 describe-snomedct-inference-job을 사용하는 방법을 보여 줍니다.

AWS CLI

SNOMED CT 추론 작업을 설명하려면

다음 describe-snomedct-inference-job 예제에서는 지정된 작업 ID를 사용하여 요청된 추론 작업의 속성을 설명합니다.

```
aws comprehendmedical describe-snomedct-inference-job \
    --job-id "2630034166536cdb52ffa3295a1b00a7"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Comprehend Medical 개발자 안내서의 온톨로지 연결 배치 분석을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeSnomedctInferenceJob의 섹션을 참조하세요. AWS CLI

### detect-entities-v2

다음 코드 예시에서는 detect-entities-v2을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 텍스트에서 직접 엔터티를 감지하려면

다음 detect-entities-v2 예제에서는 감지된 엔터티를 보여주고 입력 텍스트에서 직접 유형에 따라 레이블을 지정합니다.

```
aws comprehendmedical detect-entities-v2 \
    --text "Sleeping trouble on present dosage of Clonidine. Severe rash on face and
 leg, slightly itchy."
```

출력:

```
{
    "Id": 0,
    "BeginOffset": 38,
    "EndOffset": 47,
    "Score": 0.9942955374717712,
    "Text": "Clonidine",
    "Category": "MEDICATION",
    "Type": "GENERIC_NAME",
    "Traits": []
}
```

자세한 내용은 Amazon Comprehend Medical 개발자 안내서의 Detect Entities 버전 2를 참조하세요.

예제 2: 파일 경로에서 엔터티 감지

다음 detect-entities-v2 예제에서는 감지된 엔터티를 보여주고 파일 경로의 유형에 따라 레이블을 지정합니다.

```
aws comprehendmedical detect-entities-v2 \
    --text file://medical_entities.txt
```

medical_entities.txt의 콘텐츠:

```
{
```

```
    "Sleeping trouble on present dosage of Clonidine. Severe rash on face and leg,
 slightly itchy."
}
```

출력:

```
{
    "Id": 0,
    "BeginOffset": 38,
    "EndOffset": 47,
    "Score": 0.9942955374717712,
    "Text": "Clonidine",
    "Category": "MEDICATION",
    "Type": "GENERIC_NAME",
    "Traits": []
}
```

자세한 내용은 Amazon Comprehend Medical 개발자 안내서의 Detect Entities 버전 2를 참조하세요.

- API 자세한 내용은 AWS CLI 명령 참조의 DetectEntitiesV2를 참조하세요.

### `detect-phi`

다음 코드 예시에서는 `detect-phi`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 텍스트에서 보호된 상태 정보(PHI)를 직접 감지하려면

다음 `detect-phi` 예제에서는 입력 텍스트에서 직접 감지된 보호된 상태 정보(PHI) 엔터티를 표시합니다.

```
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."
```

출력:

```
{
```

```
    "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"
}
```


자세한 내용은 Amazon Comprehend Medical 개발자 안내서의 감지PHI를 참조하세요.

예제 2: 파일 경로에서 직접 보호 상태 정보(PHI) 감지

다음 `detect-phi` 예제는 파일 경로에서 감지된 보호된 상태 정보(PHI) 엔터티를 보여줍니다.

```
aws comprehendmedical detect-phi \
    --text file://phi.txt
```

phi.txt의 콘텐츠:

```
"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."
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon Comprehend Medical 개발자 안내서의 감지PHI를 참조하세요.

- 자세한 API 내용은 명령 참조DetectPhi의 섹션을 참조하세요. AWS CLI

### `infer-icd10-cm`

다음 코드 예시에서는 `infer-icd10-cm`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 의학적 상태 엔터티를 감지하고 텍스트에서 ICD-10-CM Ontology로 직접 연결하려면

다음 `infer-icd10-cm` 예제에서는 감지된 의료 상태 엔터티에 레이블을 지정하고 해당 엔터티를 2019년 국제 질병 분류 임상 수정판(ICD-10-CM)의 코드와 연결합니다.

```
aws comprehendmedical infer-icd10-cm \
    --text "The patient complains of abdominal pain, has a long-standing history of
 diabetes treated with Micronase daily."
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon Comprehend Medical 개발자 안내서의 추론 ICD10-CM을 참조하세요.

예제 2: 의학적 상태 엔터티를 감지하고 파일 경로에서 ICD-10-CM Ontology에 연결하려면

다음 `infer-icd-10-cm` 예제에서는 감지된 의료 상태 엔터티에 레이블을 지정하고 해당 엔터티를 2019년 국제 질병 분류 임상 수정판(ICD-10-CM)의 코드와 연결합니다.

```
aws comprehendmedical infer-icd10-cm \
    --text file://icd10cm.txt
```

`icd10cm.txt`의 콘텐츠:

```
{
    "The patient complains of abdominal pain, has a long-standing history of
 diabetes treated with Micronase daily."
```

```
}
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon Comprehend Medical 개발자 안내서의 Infer-ICD10-CM을 참조하세요.

- API 자세한 내용은 AWS CLI 명령 참조의 InferIcd10Cm을 참조하세요.

### infer-rx-norm

다음 코드 예시에서는 `infer-rx-norm`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 약물 엔터티를 감지하고 텍스트에서 에 RxNorm 직접 연결

다음 `infer-rx-norm` 예제에서는 감지된 약물 엔터티를 표시하고 레이블을 지정하며 해당 엔터티를 National Library of Medicine RxNorm 데이터베이스의 개념 식별자(RxCUI)에 연결합니다.

```
aws comprehendmedical infer-rx-norm \
    --text "Patient reports taking Levothyroxine 125 micrograms p.o. once daily, but denies taking Synthroid."
```

출력:

```
{
    "Entities": [
        {
            "Id": 0,
            "Text": "Levothyroxine",
            "Category": "MEDICATION",
            "Type": "GENERIC_NAME",
            "Score": 0.9996285438537598,
            "BeginOffset": 23,
            "EndOffset": 36,
            "Attributes": [
```

```
                {
                    "Type": "DOSAGE",
                    "Score": 0.9892290830612183,
                    "RelationshipScore": 0.9997978806495667,
                    "Id": 1,
                    "BeginOffset": 37,
                    "EndOffset": 51,
                    "Text": "125 micrograms",
                    "Traits": []
                },
                {
                    "Type": "ROUTE_OR_MODE",
                    "Score": 0.9988924860954285,
                    "RelationshipScore": 0.998291552066803,
                    "Id": 2,
                    "BeginOffset": 52,
                    "EndOffset": 56,
                    "Text": "p.o.",
                    "Traits": []
                },
                {
                    "Type": "FREQUENCY",
                    "Score": 0.9953463673591614,
                    "RelationshipScore": 0.9999889135360718,
                    "Id": 3,
                    "BeginOffset": 57,
                    "EndOffset": 67,
                    "Text": "once daily",
                    "Traits": []
                }
            ],
            "Traits": [],
            "RxNormConcepts": [
                {
                    "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet",
                    "Code": "966224",
                    "Score": 0.9912070631980896
                },
                {
                    "Description": "Levothyroxine Sodium 0.125 MG Oral Capsule",
                    "Code": "966405",
                    "Score": 0.8698278665542603
                },
                {
```

```
                    "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet
[Synthroid]",
                    "Code": "966191",
                    "Score": 0.7448257803916931
                },
                {
                    "Description": "levothyroxine",
                    "Code": "10582",
                    "Score": 0.7050482630729675
                },
                {
                    "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet
[Levoxyl]",
                    "Code": "966190",
                    "Score": 0.6921631693840027
                }
            ]
        },
        {
            "Id": 4,
            "Text": "Synthroid",
            "Category": "MEDICATION",
            "Type": "BRAND_NAME",
            "Score": 0.9946461319923401,
            "BeginOffset": 86,
            "EndOffset": 95,
            "Attributes": [],
            "Traits": [
                {
                    "Name": "NEGATION",
                    "Score": 0.5167351961135864
                }
            ],
            "RxNormConcepts": [
                {
                    "Description": "Synthroid",
                    "Code": "224920",
                    "Score": 0.9462039470672607
                },
                {
                    "Description": "Levothyroxine Sodium 0.088 MG Oral Tablet
[Synthroid]",
                    "Code": "966282",
                    "Score": 0.8309829235076904
```

```
                },
                {
                    "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet
 [Synthroid]",
                    "Code": "966191",
                    "Score": 0.4945160448551178
                },
                {
                    "Description": "Levothyroxine Sodium 0.05 MG Oral Tablet
 [Synthroid]",
                    "Code": "966247",
                    "Score": 0.3674522042274475
                },
                {
                    "Description": "Levothyroxine Sodium 0.025 MG Oral Tablet
 [Synthroid]",
                    "Code": "966158",
                    "Score": 0.2588822841644287
                }
            ]
        }
    ],
    "ModelVersion": "0.0.0"
}
```

자세한 내용은 Amazon Comprehend Medical 개발자 안내서의 추론 RxNorm을 참조하세요.

예제 2: 파일 경로 RxNorm 에서 약물 개체 및 에 대한 링크를 감지합니다.

다음 `infer-rx-norm` 예제에서는 감지된 약물 엔터티를 표시하고 레이블을 지정하며 해당 엔터티를 National Library of Medicine RxNorm 데이터베이스의 개념 식별자(RxCUI)에 연결합니다.

```
aws comprehendmedical infer-rx-norm \
    --text file://rxnorm.txt
```

`rxnorm.txt`의 콘텐츠:

```
{
    "Patient reports taking Levothyroxine 125 micrograms p.o. once daily, but denies
 taking Synthroid."
}
```

출력:

```
{
    "Entities": [
        {
            "Id": 0,
            "Text": "Levothyroxine",
            "Category": "MEDICATION",
            "Type": "GENERIC_NAME",
            "Score": 0.9996285438537598,
            "BeginOffset": 23,
            "EndOffset": 36,
            "Attributes": [
                {
                    "Type": "DOSAGE",
                    "Score": 0.9892290830612183,
                    "RelationshipScore": 0.9997978806495667,
                    "Id": 1,
                    "BeginOffset": 37,
                    "EndOffset": 51,
                    "Text": "125 micrograms",
                    "Traits": []
                },
                {
                    "Type": "ROUTE_OR_MODE",
                    "Score": 0.9988924860954285,
                    "RelationshipScore": 0.998291552066803,
                    "Id": 2,
                    "BeginOffset": 52,
                    "EndOffset": 56,
                    "Text": "p.o.",
                    "Traits": []
                },
                {
                    "Type": "FREQUENCY",
                    "Score": 0.9953463673591614,
                    "RelationshipScore": 0.9999889135360718,
                    "Id": 3,
                    "BeginOffset": 57,
                    "EndOffset": 67,
                    "Text": "once daily",
                    "Traits": []
                }
            ],
            "Traits": [],
```

```
            "RxNormConcepts": [
                {
                    "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet",
                    "Code": "966224",
                    "Score": 0.9912070631980896
                },
                {
                    "Description": "Levothyroxine Sodium 0.125 MG Oral Capsule",
                    "Code": "966405",
                    "Score": 0.8698278665542603
                },
                {
                    "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet
[Synthroid]",
                    "Code": "966191",
                    "Score": 0.7448257803916931
                },
                {
                    "Description": "levothyroxine",
                    "Code": "10582",
                    "Score": 0.7050482630729675
                },
                {
                    "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet
[Levoxyl]",
                    "Code": "966190",
                    "Score": 0.6921631693840027
                }
            ]
        },
        {
            "Id": 4,
            "Text": "Synthroid",
            "Category": "MEDICATION",
            "Type": "BRAND_NAME",
            "Score": 0.9946461319923401,
            "BeginOffset": 86,
            "EndOffset": 95,
            "Attributes": [],
            "Traits": [
                {
                    "Name": "NEGATION",
                    "Score": 0.5167351961135864
                }
```

```
            ],
            "RxNormConcepts": [
                {
                    "Description": "Synthroid",
                    "Code": "224920",
                    "Score": 0.9462039470672607
                },
                {
                    "Description": "Levothyroxine Sodium 0.088 MG Oral Tablet
 [Synthroid]",
                    "Code": "966282",
                    "Score": 0.8309829235076904
                },
                {
                    "Description": "Levothyroxine Sodium 0.125 MG Oral Tablet
 [Synthroid]",
                    "Code": "966191",
                    "Score": 0.4945160448551178
                },
                {
                    "Description": "Levothyroxine Sodium 0.05 MG Oral Tablet
 [Synthroid]",
                    "Code": "966247",
                    "Score": 0.3674522042274475
                },
                {
                    "Description": "Levothyroxine Sodium 0.025 MG Oral Tablet
 [Synthroid]",
                    "Code": "966158",
                    "Score": 0.2588822841644287
                }
            ]
        }
    ],
    "ModelVersion": "0.0.0"
}
```

자세한 내용은 Amazon Comprehend Medical 개발자 안내서의 추론 RxNorm을 참조하세요.

- 자세한 API 내용은 명령 참조InferRxNorm의 섹션을 참조하세요. AWS CLI

### infer-snomedct

다음 코드 예시에서는 infer-snomedct을 사용하는 방법을 보여 줍니다.

AWS CLI

예: 엔터티를 감지하고 텍스트에서 SNOMED CT Ontology로 직접 연결하려면

다음 infer-snomedct 예제에서는 의료 엔터티를 감지하고 2021년 3월 버전의 체계화된 의학 명명법, 임상 용어(SNOMED CT)의 개념에 연결하는 방법을 보여줍니다.

```
aws comprehendmedical infer-snomedct \
    --text "The patient complains of abdominal pain, has a long-standing history of
 diabetes treated with Micronase daily."
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon Comprehend Medical 개발자 안내서의 추론SNOMEDCT을 참조하세요.

- 자세한 API 내용은 명령 참조InferSnomedct의 섹션을 참조하세요. AWS CLI

### `list-entities-detection-v2-jobs`

다음 코드 예시에서는 `list-entities-detection-v2-jobs`을 사용하는 방법을 보여 줍니다.

AWS CLI

개체 감지 작업을 나열하려면

다음 `list-entities-detection-v2-jobs` 예제에서는 현재 비동기 감지 작업을 나열합니다.

```
aws comprehendmedical list-entities-detection-v2-jobs
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Comprehend Medical 개발자 안내서의 배치APIs를 참조하세요.

- API 자세한 내용은 AWS CLI 명령 참조의 ListEntitiesDetectionV2Jobs를 참조하세요.

### `list-icd10-cm-inference-jobs`

다음 코드 예시에서는 `list-icd10-cm-inference-jobs`을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 ICD-10-CM 추론 작업을 모두 나열하려면

다음 예제에서는 `list-icd10-cm-inference-jobs` 작업이 현재 비동기 ICD-10-CM 배치 추론 작업 목록을 반환하는 방법을 보여줍니다.

```
aws comprehendmedical list-icd10-cm-inference-jobs
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Comprehend Medical 개발자 안내서의 온톨로지 연결 배치 분석을 참조하세요.

- API 자세한 내용은 AWS CLI 명령 참조의 ListIcd10CmInferenceJobs을 참조하세요.

### `list-phi-detection-jobs`

다음 코드 예시에서는 `list-phi-detection-jobs`을 사용하는 방법을 보여 줍니다.

AWS CLI

보호된 상태 정보(PHI) 감지 작업을 나열하려면

다음 `list-phi-detection-jobs` 예제에서는 현재 보호되는 상태 정보(PHI) 감지 작업을 나열합니다.

```
aws comprehendmedical list-phi-detection-jobs
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Comprehend Medical 개발자 안내서의 배치APIs를 참조하세요.

- 자세한 API 내용은 명령 참조ListPhiDetectionJobs의 섹션을 참조하세요. AWS CLI

### `list-rx-norm-inference-jobs`

다음 코드 예시에서는 `list-rx-norm-inference-jobs`을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 Rx-Norm 추론 작업을 모두 나열하려면

다음 예제에서는 가 현재 비동기 Rx-Norm 배치 추론 작업 목록을 `list-rx-norm-inference-jobs` 반환하는 방법을 보여줍니다.

```
aws comprehendmedical list-rx-norm-inference-jobs
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Comprehend Medical 개발자 안내서의 온톨로지 연결 배치 분석을 참조하세요.

- 자세한 API 내용은 명령 참조ListRxNormInferenceJobs의 섹션을 참조하세요. AWS CLI

### `list-snomedct-inference-jobs`

다음 코드 예시에서는 `list-snomedct-inference-jobs`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 SNOMED CT 추론 작업을 나열하려면

다음 예제에서는 `list-snomedct-inference-jobs` 작업이 현재 비동기 SNOMED CT 배치 추론 작업 목록을 반환하는 방법을 보여줍니다.

```
aws comprehendmedical list-snomedct-inference-jobs
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Comprehend Medical 개발자 안내서의 온톨로지 연결 배치 분석을 참조하세요.

- 자세한 API 내용은 명령 참조ListSnomedctInferenceJobs의 섹션을 참조하세요. AWS CLI

### `start-entities-detection-v2-job`

다음 코드 예시에서는 `start-entities-detection-v2-job`을 사용하는 방법을 보여 줍니다.

AWS CLI

엔터티 감지 작업을 시작하려면

다음 `start-entities-detection-v2-job` 예제에서는 비동기 엔터티 감지 작업을 시작합니다.

```
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
```

출력:

```
{
    "JobId": "ab9887877365fe70299089371c043b96"
}
```

자세한 내용은 Amazon Comprehend Medical 개발자 안내서의 배치APIs를 참조하세요.

- API 자세한 내용은 AWS CLI 명령 참조의 StartEntitiesDetectionV2Job을 참조하세요.

### `start-icd10-cm-inference-job`

다음 코드 예시에서는 `start-icd10-cm-inference-job`을 사용하는 방법을 보여 줍니다.

AWS CLI

ICD-10-CM 추론 작업을 시작하려면

다음 `start-icd10-cm-inference-job` 예제에서는 ICD-10-CM 추론 배치 분석 작업을 시작합니다.

```
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
```

출력:

```
{
    "JobId": "ef7289877365fc70299089371c043b96"
}
```

자세한 내용은 Amazon Comprehend Medical 개발자 안내서의 Ontology 연결 배치 분석을 참조하세요.

- API 자세한 내용은 AWS CLI 명령 참조의 StartIcd10CmInferenceJob을 참조하세요.

### `start-phi-detection-job`

다음 코드 예시에서는 `start-phi-detection-job`을 사용하는 방법을 보여 줍니다.

AWS CLI

PHI 감지 작업을 시작하려면

다음 `start-phi-detection-job` 예제에서는 비동기 PHI엔터티 감지 작업을 시작합니다.

```
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
```

출력:

```
{
    "JobId": "ab9887877365fe70299089371c043b96"
}
```

자세한 내용은 Amazon Comprehend Medical 개발자 안내서의 배치APIs를 참조하세요.

- 자세한 API 내용은 명령 참조StartPhiDetectionJob의 섹션을 참조하세요. AWS CLI

### `start-rx-norm-inference-job`

다음 코드 예시에서는 `start-rx-norm-inference-job`을 사용하는 방법을 보여 줍니다.

AWS CLI

RxNorm 추론 작업을 시작하려면

다음 `start-rx-norm-inference-job` 예제에서는 RxNorm 추론 배치 분석 작업을 시작합니다.

```
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
```

출력:

```
{
    "JobId": "eg8199877365fc70299089371c043b96"
}
```

자세한 내용은 Amazon Comprehend Medical 개발자 안내서의 온톨로지 연결 배치 분석을 참조하세요.

- 자세한 API 내용은 명령 참조StartRxNormInferenceJob의 섹션을 참조하세요. AWS CLI

### `start-snomedct-inference-job`

다음 코드 예시에서는 `start-snomedct-inference-job`을 사용하는 방법을 보여 줍니다.

AWS CLI

SNOMED CT 추론 작업을 시작하려면

다음 `start-snomedct-inference-job` 예제에서는 SNOMED CT 추론 배치 분석 작업을 시작합니다.

```
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
```

출력:

```
{
    "JobId": "dg7289877365fc70299089371c043b96"
}
```

자세한 내용은 Amazon Comprehend Medical 개발자 안내서의 온톨로지 연결 배치 분석을 참조하세요.

- 자세한 API 내용은 명령 참조StartSnomedctInferenceJob의 섹션을 참조하세요. AWS CLI

### stop-entities-detection-v2-job

다음 코드 예시에서는 stop-entities-detection-v2-job을 사용하는 방법을 보여 줍니다.

AWS CLI

개체 감지 작업을 중지하려면

다음 stop-entities-detection-v2-job 예제에서는 비동기 엔터티 감지 작업을 중지합니다.

```
aws comprehendmedical stop-entities-detection-v2-job \
    --job-id "ab9887877365fe70299089371c043b96"
```

출력:

```
{
    "JobId": "ab9887877365fe70299089371c043b96"
}
```

자세한 내용은 Amazon Comprehend Medical 개발자 안내서의 배치APIs를 참조하세요.

- API 자세한 내용은 AWS CLI 명령 참조의 StopEntitiesDetectionV2Job을 참조하세요.

### stop-icd10-cm-inference-job

다음 코드 예시에서는 stop-icd10-cm-inference-job을 사용하는 방법을 보여 줍니다.

AWS CLI

ICD-10-CM 추론 작업을 중지하려면

다음 stop-icd10-cm-inference-job 예제에서는 ICD-10-CM 추론 배치 분석 작업을 중지합니다.

```
aws comprehendmedical stop-icd10-cm-inference-job \
    --job-id "4750034166536cdb52ffa3295a1b00a3"
```

출력:

```
{
    "JobId": "ef7289877365fc70299089371c043b96",
}
```

자세한 내용은 Amazon Comprehend Medical 개발자 안내서의 온톨로지 연결 배치 분석을 참조하세요.

- API 자세한 내용은 AWS CLI 명령 참조의 StopIcd10CmInferenceJob을 참조하세요.

### stop-phi-detection-job

다음 코드 예시에서는 stop-phi-detection-job을 사용하는 방법을 보여 줍니다.

AWS CLI

보호된 상태 정보(PHI) 감지 작업을 중지하려면

다음 stop-phi-detection-job 예제에서는 비동기식 보호된 상태 정보(PHI) 감지 작업을 중지합니다.

```
aws comprehendmedical stop-phi-detection-job \
    --job-id "4750034166536cdb52ffa3295a1b00a3"
```

출력:

```
{
    "JobId": "ab9887877365fe70299089371c043b96"
}
```

자세한 내용은 Amazon Comprehend Medical 개발자 안내서의 배치APIs를 참조하세요.

- 자세한 API 내용은 명령 참조StopPhiDetectionJob의 섹션을 참조하세요. AWS CLI

### stop-rx-norm-inference-job

다음 코드 예시에서는 stop-rx-norm-inference-job을 사용하는 방법을 보여 줍니다.

AWS CLI

RxNorm 추론 작업을 중지하려면

다음 stop-rx-norm-inference-job 예제에서는 ICD-10-CM 추론 배치 분석 작업을 중지합니다.

```
aws comprehendmedical stop-rx-norm-inference-job \
    --job-id "eg8199877365fc70299089371c043b96"
```

출력:

```
{
    "JobId": "eg8199877365fc70299089371c043b96",
}
```

자세한 내용은 Amazon Comprehend Medical 개발자 안내서의 온톨로지 연결 배치 분석을 참조하세요.

- 자세한 API 내용은 명령 참조StopRxNormInferenceJob의 섹션을 참조하세요. AWS CLI

### stop-snomedct-inference-job

다음 코드 예시에서는 stop-snomedct-inference-job을 사용하는 방법을 보여 줍니다.

AWS CLI

SNOMED CT 추론 작업을 중지하려면

다음 stop-snomedct-inference-job 예제에서는 SNOMED CT 추론 배치 분석 작업을 중지합니다.

```
aws comprehendmedical stop-snomedct-inference-job \
    --job-id "8750034166436cdb52ffa3295a1b00a1"
```

출력:

```
{
    "JobId": "8750034166436cdb52ffa3295a1b00a1",
}
```

자세한 내용은 Amazon Comprehend Medical 개발자 안내서의 온톨로지 연결 배치 분석을 참조하세요.

- 자세한 API 내용은 명령 참조StopSnomedctInferenceJob의 섹션을 참조하세요. AWS CLI

## AWS Config 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS Config.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `delete-config-rule`

다음 코드 예시에서는 `delete-config-rule`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS Config 규칙을 삭제하려면

다음 명령은 라는 AWS Config 규칙을 삭제합니다`MyConfigRule`.

```
aws configservice delete-config-rule --config-rule-name MyConfigRule
```

- 자세한 API 내용은 명령 참조DeleteConfigRule의 섹션을 참조하세요. AWS CLI

### delete-delivery-channel

다음 코드 예시에서는 delete-delivery-channel을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 채널을 삭제하려면

다음 명령은 기본 전송 채널을 삭제합니다.

```
aws configservice delete-delivery-channel --delivery-channel-name default
```

- 자세한 API 내용은 명령 참조DeleteDeliveryChannel의 섹션을 참조하세요. AWS CLI

### delete-evaluation-results

다음 코드 예시에서는 delete-evaluation-results을 사용하는 방법을 보여 줍니다.

AWS CLI

평가 결과를 수동으로 삭제하려면

다음 명령은 AWS 관리형 규칙 s3-에 대한 현재 평가 결과를 삭제합니다bucket-versioning-enabled.

```
aws configservice delete-evaluation-results --config-rule-name s3-bucket-versioning-
enabled
```

- 자세한 API 내용은 명령 참조DeleteEvaluationResults의 섹션을 참조하세요. AWS CLI

### deliver-config-snapshot

다음 코드 예시에서는 deliver-config-snapshot을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 스냅샷을 전송하려면

다음 명령은 기본 전송 채널에 속하는 Amazon S3 버킷에 구성 스냅샷을 전달합니다.

```
aws configservice deliver-config-snapshot --delivery-channel-name default
```

출력:

```
{
    "configSnapshotId": "d0333b00-a683-44af-921e-examplefb794"
}
```

- 자세한 API 내용은 명령 참조DeliverConfigSnapshot의 섹션을 참조하세요. AWS CLI

### describe-compliance-by-config-rule

다음 코드 예시에서는 describe-compliance-by-config-rule을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS Config 규칙에 대한 규정 준수 정보를 가져오려면

다음 명령은 하나 이상의 AWS 리소스에서 위반한 각 AWS Config 규칙에 대한 규정 준수 정보를 반환합니다.

```
aws configservice describe-compliance-by-config-rule --compliance-
types NON_COMPLIANT
```

출력에서 각 CappedCount 속성의 값은 관련 규칙을 준수하지 않는 리소스 수를 나타냅니다. 예를 들어 다음 출력은 3개의 리소스가 라는 규칙을 준수하지 않음을 나타냅니다InstanceTypesAreT2micro.

출력:

```
{
    "ComplianceByConfigRules": [
        {
            "Compliance": {
                "ComplianceContributorCount": {
                    "CappedCount": 3,
                    "CapExceeded": false
                },
                "ComplianceType": "NON_COMPLIANT"
            },
            "ConfigRuleName": "InstanceTypesAreT2micro"
        },
        {
            "Compliance": {
                "ComplianceContributorCount": {
                    "CappedCount": 10,
```

```
                    "CapExceeded": false
                },
                "ComplianceType": "NON_COMPLIANT"
            },
            "ConfigRuleName": "RequiredTagsForVolumes"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeComplianceByConfigRule의 섹션을 참조하세요. AWS CLI

### describe-compliance-by-resource

다음 코드 예시에서는 describe-compliance-by-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 리소스에 대한 규정 준수 정보를 가져오려면

다음 명령은 AWS Config에 의해 기록되고 하나 이상의 규칙을 위반하는 각 EC2 인스턴스에 대한 규정 준수 정보를 반환합니다.

```
aws configservice describe-compliance-by-resource --resource-type AWS::EC2::Instance
 --compliance-types NON_COMPLIANT
```

출력에서 각 CappedCount 속성의 값은 리소스가 위반한 규칙의 수를 나타냅니다. 예를 들어 다음 출력은 인스턴스가 2개의 규칙을 i-1a2b3c4d 위반함을 나타냅니다.

출력:

```
{
    "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"
            }
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeComplianceByResource의 섹션을 참조하세요. AWS CLI

### `describe-config-rule-evaluation-status`

다음 코드 예시에서는 `describe-config-rule-evaluation-status`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS Config 규칙에 대한 상태 정보를 가져오려면

다음 명령은 라는 AWS Config 규칙의 상태 정보를 반환합니다`MyConfigRule`.

```
aws configservice describe-config-rule-evaluation-status --config-rule-
names MyConfigRule
```

출력:

```
{
    "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"
```

```
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeConfigRuleEvaluationStatus의 섹션을 참조하세요. AWS CLI

### `describe-config-rules`

다음 코드 예시에서는 `describe-config-rules`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS Config 규칙에 대한 세부 정보를 가져오려면

다음 명령은 이름이 인 AWS Config 규칙에 대한 세부 정보를 반환합니다`InstanceTypesAreT2micro`.

```
aws configservice describe-config-rules --config-rule-names InstanceTypesAreT2micro
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeConfigRules의 섹션을 참조하세요. AWS CLI

### `describe-configuration-recorder-status`

다음 코드 예시에서는 `describe-configuration-recorder-status`을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 레코더의 상태 정보를 가져오려면

다음 명령은 기본 구성 레코더의 상태를 반환합니다.

```
aws configservice describe-configuration-recorder-status
```

출력:

```
{
    "ConfigurationRecordersStatus": [
        {
            "name": "default",
            "lastStatus": "SUCCESS",
            "recording": true,
            "lastStatusChangeTime": 1452193834.344,
            "lastStartTime": 1441039997.819,
            "lastStopTime": 1441039992.835
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeConfigurationRecorderStatus의 섹션을 참조하세요. AWS CLI

### describe-configuration-recorders

다음 코드 예시에서는 describe-configuration-recorders을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 레코더에 대한 세부 정보를 가져오려면

다음 명령은 기본 구성 레코더에 대한 세부 정보를 반환합니다.

```
aws configservice describe-configuration-recorders
```

출력:

```
{
    "ConfigurationRecorders": [
        {
            "recordingGroup": {
                "allSupported": true,
                "resourceTypes": [],
                "includeGlobalResourceTypes": true
            },
            "roleARN": "arn:aws:iam::123456789012:role/config-ConfigRole-
A1B2C3D4E5F6",
            "name": "default"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeConfigurationRecorders의 섹션을 참조하세요. AWS CLI

### describe-delivery-channel-status

다음 코드 예시에서는 describe-delivery-channel-status을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 채널의 상태 정보를 가져오려면

다음 명령은 전송 채널의 상태를 반환합니다.

```
aws configservice describe-delivery-channel-status
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeDeliveryChannelStatus의 섹션을 참조하세요. AWS CLI

### `describe-delivery-channels`

다음 코드 예시에서는 `describe-delivery-channels`을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 채널에 대한 세부 정보를 가져오려면

다음 명령은 전송 채널에 대한 세부 정보를 반환합니다.

```
aws configservice describe-delivery-channels
```

출력:

```
{
```

```
    "DeliveryChannels": [
        {
            "snsTopicARN": "arn:aws:sns:us-east-1:123456789012:config-topic",
            "name": "default",
            "s3BucketName": "config-bucket-123456789012"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeDeliveryChannels의 섹션을 참조하세요. AWS CLI

### get-compliance-details-by-config-rule

다음 코드 예시에서는 get-compliance-details-by-config-rule을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS Config 규칙에 대한 평가 결과를 가져오려면

다음 명령은 라는 AWS Config 규칙을 준수하지 않는 모든 리소스에 대한 평가 결과를 반환합니다InstanceTypesAreT2micro.

```
aws configservice get-compliance-details-by-config-rule --config-rule-
name InstanceTypesAreT2micro --compliance-types NON_COMPLIANT
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetComplianceDetailsByConfigRule의 섹션을 참조하세요. AWS CLI

### `get-compliance-details-by-resource`

다음 코드 예시에서는 `get-compliance-details-by-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 리소스에 대한 평가 결과를 가져오려면

다음 명령은 EC2 인스턴스가 `i-1a2b3c4d` 준수하지 않는 각 규칙에 대한 평가 결과를 반환합니다.

```
aws configservice get-compliance-details-by-resource --resource-
type AWS::EC2::Instance --resource-id i-1a2b3c4d --compliance-types NON_COMPLIANT
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetComplianceDetailsByResource의 섹션을 참조하세요. AWS CLI

### get-compliance-summary-by-config-rule

다음 코드 예시에서는 get-compliance-summary-by-config-rule을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS Config 규칙에 대한 규정 준수 요약을 가져오려면

다음 명령은 규정을 준수하는 규칙 수와 규정을 준수하지 않는 규칙 수를 반환합니다.

```
aws configservice get-compliance-summary-by-config-rule
```

출력에서 각 CappedCount 속성의 값은 규정을 준수하거나 준수하지 않는 규칙의 수를 나타냅니다.

출력:

```
{
    "ComplianceSummary": {
        "NonCompliantResourceCount": {
            "CappedCount": 3,
            "CapExceeded": false
        },
        "ComplianceSummaryTimestamp": 1452204131.493,
        "CompliantResourceCount": {
            "CappedCount": 2,
            "CapExceeded": false
        }
    }
}
```

- 자세한 API 내용은 명령 참조GetComplianceSummaryByConfigRule의 섹션을 참조하세요. AWS CLI

### get-compliance-summary-by-resource-type

다음 코드 예시에서는 get-compliance-summary-by-resource-type을 사용하는 방법을 보여줍니다.

AWS CLI

모든 리소스 유형에 대한 규정 준수 요약을 가져오려면

다음 명령은 규정을 준수하지 않는 AWS 리소스 수와 규정을 준수하는 리소스 수를 반환합니다.

```
aws configservice get-compliance-summary-by-resource-type
```

출력에서 각 CappedCount 속성의 값은 규정 준수 또는 규정 미준수 리소스 수를 나타냅니다.

출력:

```
{
    "ComplianceSummariesByResourceType": [
        {
            "ComplianceSummary": {
                "NonCompliantResourceCount": {
                    "CappedCount": 16,
                    "CapExceeded": false
                },
                "ComplianceSummaryTimestamp": 1453237464.543,
                "CompliantResourceCount": {
                    "CappedCount": 10,
                    "CapExceeded": false
                }
            }
        }
    ]
}
```

특정 리소스 유형에 대한 규정 준수 요약을 가져오려면

다음 명령은 규정을 준수하지 않는 EC2 인스턴스 수와 규정을 준수하는 인스턴스 수를 반환합니다.

```
aws configservice get-compliance-summary-by-resource-type --resource-
types AWS::EC2::Instance
```

출력에서 각 CappedCount 속성의 값은 규정 준수 또는 규정 미준수 리소스 수를 나타냅니다.

출력:

```
{
    "ComplianceSummariesByResourceType": [
        {
            "ResourceType": "AWS::EC2::Instance",
            "ComplianceSummary": {
                "NonCompliantResourceCount": {
                    "CappedCount": 3,
                    "CapExceeded": false
```

```
                },
                "ComplianceSummaryTimestamp": 1452204923.518,
                "CompliantResourceCount": {
                    "CappedCount": 7,
                    "CapExceeded": false
                }
            }
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetComplianceSummaryByResourceType의 섹션을 참조하세요. AWS CLI

### get-resource-config-history

다음 코드 예시에서는 get-resource-config-history을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 리소스의 구성 기록을 가져오려면

다음 명령은 ID가 인 EC2 인스턴스의 구성 항목 목록을 반환합니다i-1a2b3c4d.

```
aws configservice get-resource-config-history --resource-type AWS::EC2::Instance --
resource-id i-1a2b3c4d
```

- 자세한 API 내용은 명령 참조GetResourceConfigHistory의 섹션을 참조하세요. AWS CLI

### get-status

다음 코드 예시에서는 get-status을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS Config 상태를 가져오려면

다음 명령은 전송 채널 및 구성 레코더의 상태를 반환합니다.

```
aws configservice get-status
```

출력:

```
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
```

- 자세한 API 내용은 명령 참조GetStatus의 섹션을 참조하세요. AWS CLI

### `list-discovered-resources`

다음 코드 예시에서는 `list-discovered-resources`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS Config가 검색한 리소스를 나열하려면

다음 명령은 AWS Config가 발견한 EC2 인스턴스를 나열합니다.

```
aws configservice list-discovered-resources --resource-type AWS::EC2::Instance
```

출력:

```
{
    "resourceIdentifiers": [
        {
            "resourceType": "AWS::EC2::Instance",
            "resourceId": "i-1a2b3c4d"
        },
        {
            "resourceType": "AWS::EC2::Instance",
            "resourceId": "i-2a2b3c4d"
        },
        {
```

```
            "resourceType": "AWS::EC2::Instance",
            "resourceId": "i-3a2b3c4d"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListDiscoveredResources의 섹션을 참조하세요. AWS CLI

### put-config-rule

다음 코드 예시에서는 put-config-rule을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 관리형 Config 규칙을 추가하려면

다음 명령은 AWS 관리형 Config 규칙을 추가하는 JSON 코드를 제공합니다.

```
aws configservice put-config-rule --config-rule file://
RequiredTagsForEC2Instances.json
```

RequiredTagsForEC2Instances.json 는 규칙 구성을 포함하는 JSON 파일입니다.

```
{
  "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\"}"
}
```

ComplianceResourceTypes 속성의 경우 이 JSON 코드는 범위를 AWS::EC2::Instance 유형의 리소스로 제한하므로 AWS Config는 규칙에 대해 EC2 인스턴스만 평가합니다. 규칙은

관리형 규칙이므로 Owner 속성은 AWS로 설정되고 SourceIdentifier 속성은 규칙 식별자인 REQUIRED_TAGS로 설정됩니다. InputParameters 속성의 경우 규칙에 필요한 태그 키인 CostCenter 및 Owner가 지정됩니다.

명령이 성공하면 AWS Config는 출력을 반환하지 않습니다. 규칙 구성을 확인하려면 명령을 실행 describe-config-rules하고 규칙 이름을 지정합니다.

고객 관리형 Config 규칙을 추가하는 방법

다음 명령은 고객 관리형 Config 규칙을 추가하는 JSON 코드를 제공합니다.

```
aws configservice put-config-rule --config-rule file://InstanceTypesAreT2micro.json
```

InstanceTypesAreT2micro.json 는 규칙 구성을 포함하는 JSON 파일입니다.

```
{
  "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\"}"
}
```

ComplianceResourceTypes 속성의 경우 이 JSON 코드는 범위를 AWS::EC2::Instance 유형의 리소스로 제한하므로 AWS Config는 규칙에 대해 EC2 인스턴스만 평가합니다. 이 규칙은 고객 관리형 규칙이므로 Owner 속성은 로 설정CUSTOM_LAMBDA되고 SourceIdentifier 속성은 AWS Lambda 함수ARN의 로 설정됩니다. SourceDetails 객체가 필요합니다.

InputParameters 속성에 지정된 파라미터는 AWS Config가 규칙을 기준으로 리소스를 평가하기 위해 호출할 때 AWS Lambda 함수로 전달됩니다.

명령이 성공하면 AWS Config는 출력을 반환하지 않습니다. 규칙 구성을 확인하려면 명령을 실행 describe-config-rules하고 규칙 이름을 지정합니다.

- 자세한 API 내용은 명령 참조PutConfigRule의 섹션을 참조하세요. AWS CLI

### put-configuration-recorder

다음 코드 예시에서는 put-configuration-recorder을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 지원되는 모든 리소스를 기록하려면

다음 명령은 글로벌 리소스 유형을 포함하여 지원되는 모든 리소스 유형에 대한 변경 사항을 추적하는 구성 레코더를 생성합니다.

```
aws configservice put-configuration-recorder \
    --configuration-recorder name=default,roleARN=arn:aws:iam::123456789012:role/config-role \
    --recording-group allSupported=true,includeGlobalResourceTypes=true
```

명령이 성공하면 AWS Config는 출력을 반환하지 않습니다. 구성 레코더의 설정을 확인하려면 명령을 실행합니다 describe-configuration-recorders.

예제 2: 특정 유형의 리소스를 기록하려면

다음 명령은 --recording-group 옵션의 JSON 파일에 지정된 리소스 유형에 대한 변경 사항만 추적하는 구성 레코더를 생성합니다.

```
aws configservice put-configuration-recorder \
    --configuration-recorder name=default,roleARN=arn:aws:iam::123456789012:role/config-role \
    --recording-group file://recordingGroup.json
```

recordingGroup.json은 AWS Config가 기록할 리소스 유형을 지정하는 JSON 파일입니다.

```
{
    "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"
    ]
}
```

resourceTypes 키에 대한 리소스 유형을 지정하려면 먼저 allSupported 및 includeGlobalResource 유형 옵션을 거짓 또는 생략하도록 설정해야 합니다.

명령이 성공하면 AWS Config는 출력을 반환하지 않습니다. 구성 레코더의 설정을 확인하려면 명령을 실행합니다 describe-configuration-recorders.

예제 3: 특정 유형의 리소스를 제외한 지원되는 모든 리소스를 선택하려면

다음 명령은 --recording-group 옵션의 JSON 파일에 지정된 리소스 유형을 제외한 현재 및 향후 지원되는 모든 리소스 유형에 대한 변경 사항을 추적하는 구성 레코더를 생성합니다.

```
aws configservice put-configuration-recorder \
    --configuration-recorder name=default,roleARN=arn:aws:iam::123456789012:role/
config-role \
    --recording-group file://recordingGroup.json
```

recordingGroup.json은 AWS Config가 기록할 리소스 유형을 지정하는 JSON 파일입니다.

```
{
    "allSupported": false,
    "exclusionByResourceTypes": {
        "resourceTypes": [
        "AWS::Redshift::ClusterSnapshot",
        "AWS::RDS::DBClusterSnapshot",
        "AWS::CloudFront::StreamingDistribution"
    ]
    },
        "includeGlobalResourceTypes": false,
        "recordingStrategy": {
```

```
        "useOnly": "EXCLUSION_BY_RESOURCE_TYPES"
    },
}
```

레코딩에서 제외할 리소스 유형을 지정하려면 1) allSupported 및 includeGlobalResource유형 옵션을 거짓 또는 생략하도록 설정하고 2) 의 useOnly 필드를 EXCLUSION_BY_RESOURCE_ RecordingStrategy 로 설정해야 합니다TYPES.

명령이 성공하면 AWS Config는 출력을 반환하지 않습니다. 구성 레코더의 설정을 확인하려면 명령을 실행합니다 describe-configuration-recorders.

- 자세한 API 내용은 명령 참조PutConfigurationRecorder의 섹션을 참조하세요. AWS CLI

### put-delivery-channel

다음 코드 예시에서는 put-delivery-channel을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 채널을 생성하려면

다음 명령은 전송 채널에 대한 설정을 JSON 코드로 제공합니다.

```
aws configservice put-delivery-channel --delivery-channel file://
deliveryChannel.json
```

deliveryChannel.json 파일은 전송 채널 속성을 지정합니다.

```
{
    "name": "default",
    "s3BucketName": "config-bucket-123456789012",
    "snsTopicARN": "arn:aws:sns:us-east-1:123456789012:config-topic",
    "configSnapshotDeliveryProperties": {
        "deliveryFrequency": "Twelve_Hours"
    }
}
```

이 예에서는 다음 속성을 설정합니다.

name - 전송 채널의 이름입니다. 기본적으로 AWS Config는 default 새 전송 채널에 이름을 할당합니다. put-delivery-channel 명령을 사용하여 전송 채널 이름을 업데이트할 수 없습니다. 이

름을 변경하는 단계는 전송 채널 이름 바꾸기를 참조하세요.s3BucketName - AWS Config가 구성 스냅샷 및 구성 기록 파일을 전달하는 Amazon S3 버킷의 이름입니다. 다른 AWS 계정에 속하는 버킷을 지정하는 경우 해당 버킷에는 AWS Config에 액세스 권한을 부여하는 정책이 있어야 합니다. 자세한 내용을 알아보려면 Amazon S3 버킷에 대한 권한을 참조하세요.

snsTopicARN - AWS Config가 구성 변경에 대한 알림을 보내는 Amazon SNS 주제의 Amazon 리소스 이름(ARN)입니다. 다른 계정에서 주제를 선택하면 주제에 AWS Config에 액세스 권한을 부여하는 정책이 있어야 합니다. 자세한 내용은 Amazon SNS 주제에 대한 권한을 참조하세요.

configSnapshotDeliveryProperties - AWS Config가 구성 스냅샷을 전송하는 빈도와 주기적 Config 규칙에 대한 평가를 호출하는 빈도를 설정하는 deliveryFrequency 속성을 포함합니다.

명령이 성공하면 AWS Config는 출력을 반환하지 않습니다. 전송 채널의 설정을 확인하려면 명령을 실행합니다 describe-delivery-channels.

- 자세한 API 내용은 명령 참조PutDeliveryChannel의 섹션을 참조하세요. AWS CLI

### start-config-rules-evaluation

다음 코드 예시에서는 start-config-rules-evaluation을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS Config 규칙에 대한 온디맨드 평가를 실행하려면

다음 명령은 두 가지 AWS 관리형 규칙에 대한 평가를 시작합니다.

```
aws configservice start-config-rules-evaluation --config-rule-names s3-bucket-versioning-enabled cloudtrail-enabled
```

- 자세한 API 내용은 명령 참조StartConfigRulesEvaluation의 섹션을 참조하세요. AWS CLI

### start-configuration-recorder

다음 코드 예시에서는 start-configuration-recorder을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 레코더를 시작하려면

다음 명령은 기본 구성 레코더를 시작합니다.

```
aws configservice start-configuration-recorder --configuration-recorder-name default
```

명령이 성공하면 AWS Config는 출력을 반환하지 않습니다. AWS Config가 리소스를 기록하고 있는지 확인하려면 get-status 명령을 실행합니다.

- 자세한 API 내용은 명령 참조StartConfigurationRecorder의 섹션을 참조하세요. AWS CLI

### stop-configuration-recorder

다음 코드 예시에서는 stop-configuration-recorder을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 레코더를 중지하려면

다음 명령은 기본 구성 레코더를 중지합니다.

```
aws configservice stop-configuration-recorder --configuration-recorder-name default
```

명령이 성공하면 AWS Config는 출력을 반환하지 않습니다. AWS Config가 리소스를 기록하지 않는지 확인하려면 get-status 명령을 실행합니다.

- 자세한 API 내용은 명령 참조StopConfigurationRecorder의 섹션을 참조하세요. AWS CLI

### subscribe

다음 코드 예시에서는 subscribe을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS Config를 구독하려면

다음 명령은 기본 전송 채널 및 구성 레코더를 생성합니다. 또한 명령은 AWS Config가 구성 정보를 전달할 Amazon S3 버킷 및 Amazon SNS 주제를 지정합니다.

```
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
```

출력:

```
Using existing S3 bucket: config-bucket-123456789012
Using existing SNS topic: arn:aws:sns:us-east-1:123456789012:config-topic
Subscribe succeeded:

Configuration Recorders: [
    {
        "recordingGroup": {
            "allSupported": true,
            "resourceTypes": [],
            "includeGlobalResourceTypes": false
        },
        "roleARN": "arn:aws:iam::123456789012:role/ConfigRole-A1B2C3D4E5F6",
        "name": "default"
    }
]

Delivery Channels: [
    {
        "snsTopicARN": "arn:aws:sns:us-east-1:123456789012:config-topic",
        "name": "default",
        "s3BucketName": "config-bucket-123456789012"
    }
]
```

- API 자세한 내용은 AWS CLI 명령 참조의 구독을 참조하세요.

## 를 사용한 Amazon Connect 예제 AWS CLI

다음 코드 예제에서는 Amazon Connect AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### create-user

다음 코드 예시에서는 create-user을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자를 생성하려면

다음 create-user 예제에서는 지정된 속성을 가진 사용자를 지정된 Amazon Connect 인스턴스에 추가합니다.

```
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
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon Connect 관리자 안내서의 사용자 추가를 참조하세요.

- 자세한 API 내용은 명령 참조CreateUser의 섹션을 참조하세요. AWS CLI

### delete-user

다음 코드 예시에서는 delete-user을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 삭제

다음 delete-user 예제에서는 지정된 Amazon Connect 인스턴스에서 지정된 사용자를 삭제합니다.

```
aws connect delete-user \
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --user-id 87654321-2222-1234-1234-111234567891
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Connect 관리자 안내서의 사용자 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteUser의 섹션을 참조하세요. AWS CLI

### describe-user-hierarchy-group

다음 코드 예시에서는 describe-user-hierarchy-group을 사용하는 방법을 보여 줍니다.

AWS CLI

계층 그룹에 대한 세부 정보를 표시하려면

다음 describe-user-hierarchy-group 예제에서는 지정된 Amazon Connect 계층 구조 그룹에 대한 세부 정보를 표시합니다.

```
aws connect describe-user-hierarchy-group \
    --hierarchy-group-id 12345678-1111-2222-800e-aaabbb555gg \
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

출력:

```
{
     "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"
            }
        }
    }
}
```

자세한 내용은 Amazon Connect 관리자 안내서의 에이전트 계층 설정을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeUserHierarchyGroup의 섹션을 참조하세요. AWS CLI

### describe-user-hierarchy-structure

다음 코드 예시에서는 describe-user-hierarchy-structure을 사용하는 방법을 보여 줍니다.

AWS CLI

계층 구조에 대한 세부 정보를 표시하려면

다음 describe-user-hierarchy-structure 예제에서는 지정된 Amazon Connect 인스턴스의 계층 구조에 대한 세부 정보를 표시합니다.

```
aws connect describe-user-hierarchy-group \
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 Amazon Connect 관리자 안내서의 에이전트 계층 설정을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeUserHierarchyStructure의 섹션을 참조하세요. AWS CLI

### describe-user

다음 코드 예시에서는 describe-user을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자의 세부 정보를 표시하려면

다음 describe-user 예제에서는 지정된 Amazon Connect 사용자에 대한 세부 정보를 표시합니다.

```
aws connect describe-user \
    --user-id 0c245dc0-0cf5-4e37-800e-2a7481cc8a60
    --instance-id 40c83b68-ea62-414c-97bb-d018e39e158e
```

출력:

```
{
    "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": {}
    }
}
```

자세한 내용은 Amazon Connect 관리자 안내서의 사용자 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeUser의 섹션을 참조하세요. AWS CLI

### get-contact-attributes

다음 코드 예시에서는 get-contact-attributes을 사용하는 방법을 보여 줍니다.

AWS CLI

연락처의 속성을 검색하려면

다음 get-contact-attributes 예제에서는 지정된 Amazon Connect 연락처에 대해 설정된 속성을 검색합니다.

```
aws connect get-contact-attributes \
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --initial-contact-id 12345678-1111-2222-800e-a2b3c4d5f6g7
```

출력:

```
{
    "Attributes": {
        "greetingPlayed": "true"
```

```
    }
}
```

자세한 내용은 Amazon Connect 관리자 안내서의 Amazon Connect 연락처 속성 사용을 참조하세요. Amazon Connect

- 자세한 API 내용은 명령 참조GetContactAttributes의 섹션을 참조하세요. AWS CLI

### `list-contact-flows`

다음 코드 예시에서는 `list-contact-flows`을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스의 연락처 흐름을 나열하려면

다음 `list-contact-flows` 예제에서는 지정된 Amazon Connect 인스턴스의 고객 응대 흐름을 나열합니다.

```
aws connect list-contact-flows \
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

출력:

```
{
     "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"
        },
    ]
}
```

자세한 내용은 Amazon Connect 관리자 안내서의 Amazon Connect 고객 응대 흐름 생성을 참조하세요. Amazon Connect

- 자세한 API 내용은 명령 참조ListContactFlows의 섹션을 참조하세요. AWS CLI

### `list-hours-of-operations`

다음 코드 예시에서는 `list-hours-of-operations`을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스에서 작업 시간을 나열하려면

다음 `list-hours-of-operations` 예제에서는 지정된 Amazon Connect 인스턴스의 작업 시간을 나열합니다.

```
aws connect list-hours-of-operations \
    --instance-id 40c83b68-ea62-414c-97bb-d018e39e158e
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Connect 관리자 안내서의 대기열에 대한 작업 시간 설정을 참조하세요.

- 자세한 API 내용은 명령 참조ListHoursOfOperations의 섹션을 참조하세요. AWS CLI

### `list-phone-numbers`

다음 코드 예시에서는 `list-phone-numbers`을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스의 전화번호를 나열하려면

다음 `list-phone-numbers` 예제에서는 지정된 Amazon Connect 인스턴스의 전화번호를 나열합니다.

```
aws connect list-phone-numbers \
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Connect 관리자 안내서의 고객 센터에 대한 전화번호 설정을 참조하세요.

- 자세한 API 내용은 명령 참조ListPhoneNumbers의 섹션을 참조하세요. AWS CLI

### `list-queues`

다음 코드 예시에서는 `list-queues`을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스의 대기열을 나열하려면

다음 `list-queues` 예제에서는 지정된 Amazon Connect 인스턴스의 대기열을 나열합니다.

```
aws connect list-queues \
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

출력:

```
{
    "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"
        },
    ]
}
```

자세한 내용은 Amazon Connect 관리자 안내서의 대기열 생성을 참조하세요.

- 자세한 API 내용은 명령 참조ListQueues의 섹션을 참조하세요. AWS CLI

### `list-routing-profiles`

다음 코드 예시에서는 `list-routing-profiles`을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스의 라우팅 프로파일을 나열하려면

다음 `list-routing-profiles` 예제에서는 지정된 Amazon Connect 인스턴스의 라우팅 프로파일을 나열합니다.

```
aws connect list-routing-profiles \
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

출력:

```
{
    "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"
        },
    ]
}
```

자세한 내용은 Amazon Connect 관리자 안내서의 라우팅 프로필 생성을 참조하세요.

- 자세한 API 내용은 명령 참조ListRoutingProfiles의 섹션을 참조하세요. AWS CLI

### `list-security-profiles`

다음 코드 예시에서는 `list-security-profiles`을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스의 보안 프로필을 나열하려면

다음 `list-security-profiles` 예제에서는 지정된 Amazon Connect 인스턴스의 보안 프로파일을 나열합니다.

```
aws connect list-security-profiles \
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Connect 관리자 안내서의 권한 할당: 보안 프로필을 참조하세요.

- 자세한 API 내용은 명령 참조ListSecurityProfiles의 섹션을 참조하세요. AWS CLI

### `list-user-hierarchy-groups`

다음 코드 예시에서는 `list-user-hierarchy-groups`을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스의 사용자 계층 그룹을 나열하려면

다음 `list-user-hierarchy-groups` 예제에서는 지정된 Amazon Connect 인스턴스의 사용자 계층 그룹을 나열합니다.

```
aws connect list-user-hierarchy-groups \
    --instance-id 40c83b68-ea62-414c-97bb-d018e39e158e
```

출력:

```
{
    "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"
        },
    ]
}
```

자세한 내용은 Amazon Connect 관리자 안내서의 에이전트 계층 설정을 참조하세요.

- 자세한 API 내용은 명령 참조ListUserHierarchyGroups의 섹션을 참조하세요. AWS CLI

### list-users

다음 코드 예시에서는 list-users을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스의 사용자 계층 그룹을 나열하려면

다음 list-users 예제에서는 지정된 Amazon Connect 인스턴스의 사용자를 나열합니다.

```
aws connect list-users \
    --instance-id 40c83b68-ea62-414c-97bb-d018e39e158e
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Connect 관리자 안내서의 사용자 추가를 참조하세요.

- 자세한 API 내용은 명령 참조ListUsers의 섹션을 참조하세요. AWS CLI

### update-contact-attributes

다음 코드 예시에서는 update-contact-attributes을 사용하는 방법을 보여 줍니다.

AWS CLI

연락처의 속성을 업데이트하려면

다음 update-contact-attributes 예제에서는 지정된 Amazon Connect 사용자의 greetingPlayed 속성을 업데이트합니다.

```
aws connect update-contact-attributes \
    --initial-contact-id 11111111-2222-3333-4444-12345678910 \
    --instance-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --attributes greetingPlayed=false
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Connect 관리자 안내서의 Amazon Connect 연락처 속성 사용을 참조하세요. Amazon Connect

- 자세한 API 내용은 명령 참조UpdateContactAttributes의 섹션을 참조하세요. AWS CLI

### update-user-hierarchy

다음 코드 예시에서는 update-user-hierarchy을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자의 계층을 업데이트하려면

다음 update-user-hierarchy 예제에서는 지정된 Amazon Connect 사용자의 에이전트 계층 구조를 업데이트합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Connect 관리자 안내서의 에이전트 설정 구성을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateUserHierarchy의 섹션을 참조하세요. AWS CLI

### update-user-identity-info

다음 코드 예시에서는 update-user-identity-info을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자의 자격 증명 정보를 업데이트하려면

다음 update-user-identity-info 예제에서는 지정된 Amazon Connect 사용자의 자격 증명 정보를 업데이트합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Connect 관리자 안내서의 에이전트 설정 구성을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateUserIdentityInfo의 섹션을 참조하세요. AWS CLI

### update-user-phone-config

다음 코드 예시에서는 update-user-phone-config을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자의 전화 구성을 업데이트하려면

다음 update-user-phone-config 예제에서는 지정된 사용자의 전화 구성을 업데이트합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Connect 관리자 안내서의 에이전트 설정 구성을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateUserPhoneConfig의 섹션을 참조하세요. AWS CLI

### update-user-routing-profile

다음 코드 예시에서는 update-user-routing-profile을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자의 라우팅 프로필을 업데이트하려면

다음 update-user-routing-profile 예제에서는 지정된 Amazon Connect 사용자의 라우팅 프로파일을 업데이트합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Connect 관리자 안내서의 에이전트 설정 구성을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateUserRoutingProfile의 섹션을 참조하세요. AWS CLI

### update-user-security-profiles

다음 코드 예시에서는 update-user-security-profiles을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자의 보안 프로필을 업데이트하려면

다음 update-user-security-profiles 예제에서는 지정된 Amazon Connect 사용자의 보안 프로파일을 업데이트합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Connect 관리자 안내서의 권한 할당: 보안 프로필을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateUserSecurityProfiles의 섹션을 참조하세요. AWS CLI

## AWS Cost and Usage Report 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS Cost and Usage Report.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `delete-report-definition`

다음 코드 예시에서는 `delete-report-definition`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 비용 및 사용 보고서를 삭제하려면

이 예제에서는 AWS 비용 및 사용 보고서를 삭제합니다.

명령:

```
aws cur --region us-east-1 delete-report-definition --report-name "ExampleReport"
```

- 자세한 API 내용은 명령 참조DeleteReportDefinition의 섹션을 참조하세요. AWS CLI

### describe-report-definitions

다음 코드 예시에서는 describe-report-definitions을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 비용 및 사용 보고서 목록을 검색하려면

이 예제에서는 계정이 소유한 AWS 비용 및 사용 보고서 목록을 설명합니다.

명령:

```
aws cur --region us-east-1 describe-report-definitions --max-items 5
```

출력:

```
  {
"ReportDefinitions": [
  {
      "ReportName": "ExampleReport",
      "Compression": "ZIP",
      "S3Region": "us-east-1",
      "Format": "textORcsv",
      "S3Prefix": "exampleprefix",
      "S3Bucket": "example-s3-bucket",
      "TimeUnit": "DAILY",
      "AdditionalArtifacts": [
          "REDSHIFT",
          "QUICKSIGHT"
      ],
      "AdditionalSchemaElements": [
          "RESOURCES"
      ]
  }
]
  }
```

- 자세한 API 내용은 명령 참조DescribeReportDefinitions의 섹션을 참조하세요. AWS CLI

### put-report-definition

다음 코드 예시에서는 put-report-definition을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 비용 및 사용 보고서를 생성하려면

다음 put-report-definition 예제에서는 Amazon Redshift 또는 Amazon 에 업로드할 수 있는 일일 AWS 비용 및 사용 보고서를 생성합니다 QuickSight.

```
aws cur put-report-definition --report-definition file://report-definition.json
```

report-definition.json의 콘텐츠:

```
{
    "ReportName": "ExampleReport",
    "TimeUnit": "DAILY",
    "Format": "textORcsv",
    "Compression": "ZIP",
    "AdditionalSchemaElements": [
        "RESOURCES"
    ],
    "S3Bucket": "example-s3-bucket",
    "S3Prefix": "exampleprefix",
    "S3Region": "us-east-1",
    "AdditionalArtifacts": [
        "REDSHIFT",
        "QUICKSIGHT"
    ]
}
```

- 자세한 API 내용은 명령 참조PutReportDefinition의 섹션을 참조하세요. AWS CLI

## 를 사용한 Cost Explorer 서비스 예제 AWS CLI

다음 코드 예제에서는 Cost Explorer Service와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### get-cost-and-usage

다음 코드 예시에서는 get-cost-and-usage을 사용하는 방법을 보여 줍니다.

AWS CLI

2017년 9월 한 달 동안 계정의 S3 사용량을 검색하려면

다음 get-cost-and-usage 예제에서는 2017년 9월 한 달 동안 계정의 S3 사용량을 검색합니다.

```
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
```

filters.json의 콘텐츠:

```
{
    "Dimensions": {
        "Key": "SERVICE",
        "Values": [
            "Amazon Simple Storage Service"
        ]
    }
}
```

출력:

```
{
    "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"
                        }
                    }
                }
            ]
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetCostAndUsage의 섹션을 참조하세요. AWS CLI

### `get-dimension-values`

다음 코드 예시에서는 `get-dimension-values`을 사용하는 방법을 보여 줍니다.

AWS CLI

'Elastic' SERVICE값으로 차원 에 대한 태그를 검색하려면

이 예제에서는 2017년 1월 1일부터 2017년 5월 18일까지 'Elastic' SERVICE값으로 차원 에 대한 태그를 검색합니다.

명령:

```
aws ce get-dimension-values --search-string Elastic --time-
period Start=2017-01-01,End=2017-05-18 --dimension SERVICE
```

출력:

```
{
   "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
}
```

- 자세한 API 내용은 명령 참조GetDimensionValues의 섹션을 참조하세요. AWS CLI

### `get-reservation-coverage`

다음 코드 예시에서는 `get-reservation-coverage`을 사용하는 방법을 보여 줍니다.

AWS CLI

us-east-1 리전의 EC2 t2.nano 인스턴스에 대한 예약 적용 범위를 검색하려면

이 예제는 EC2 2017년 7월~9월 us-east-1 리전의 t2.nano 인스턴스에 대한 예약 범위를 검색합니다.

명령:

```
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
```

filter.json:

```
{
   "And": [
     {
       "Dimensions": {
         "Key": "INSTANCE_TYPE",
         "Values": [
           "t2.nano"
         ]
       },
       "Dimensions": {
         "Key": "REGION",
         "Values": [
           "us-east-1"
         ]
       }
     }
   ]
 }
```

출력:

```
{
   "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
}
```

- 자세한 API 내용은 명령 참조GetReservationCoverage의 섹션을 참조하세요. AWS CLI

### get-reservation-purchase-recommendation

다음 코드 예시에서는 get-reservation-purchase-recommendation을 사용하는 방법을 보여 줍니다.

AWS CLI

3년 기간EC2RIs으로 부분 선불에 대한 예약 권장 사항을 검색하려면

다음 get-reservation-purchase-recommendation 예제에서는 지난 60일 EC2 사용량을 기준으로 3년 기간의 부분 선불 EC2 인스턴스에 대한 권장 사항을 검색합니다.

```
aws ce get-reservation-purchase-recommendation \
    --service "Amazon Redshift" \
    --lookback-period-in-days SIXTY_DAYS \
    --term-in-years THREE_YEARS \
    --payment-option PARTIAL_UPFRONT
```

출력:

```
{
    "Recommendations": [],
    "Metadata": {
        "GenerationTimestamp": "2018-08-08T15:20:57Z",
        "RecommendationId": "00d59dde-a1ad-473f-8ff2-iexample3330b"
    }
}
```

- 자세한 API 내용은 명령 참조GetReservationPurchaseRecommendation의 섹션을 참조하세요. AWS CLI

### get-reservation-utilization

다음 코드 예시에서는 get-reservation-utilization을 사용하는 방법을 보여 줍니다.

AWS CLI

계정의 예약 사용률을 검색하려면

다음 get-reservation-utilization 예제에서는 계정의 모든 t2.nano 인스턴스 유형에 대한 RI 사용률을 2018-03-01부터 2018-08-01까지 검색합니다.

```
aws ce get-reservation-utilization \
    --time-period Start=2018-03-01,End=2018-08-01 \
    --filter file://filters.json
```

filters.json의 콘텐츠:

```
{
    "Dimensions": {
        "Key": "INSTANCE_TYPE",
        "Values": [
            "t2.nano"
        ]
    }
}
```

출력:

```
{
    "Total": {
        "TotalAmortizedFee": "0",
        "UtilizationPercentage": "0",
        "PurchasedHours": "0",
        "NetRISavings": "0",
        "TotalActualHours": "0",
        "AmortizedRecurringFee": "0",
        "UnusedHours": "0",
        "TotalPotentialRISavings": "0",
        "OnDemandCostOfRIHoursUsed": "0",
        "AmortizedUpfrontFee": "0"
    },
    "UtilizationsByTime": []
```

```
}
```

- 자세한 API 내용은 명령 참조GetReservationUtilization의 섹션을 참조하세요. AWS CLI

### `get-tags`

다음 코드 예시에서는 `get-tags`을 사용하는 방법을 보여 줍니다.

AWS CLI

비용 할당 태그의 키 및 값을 검색하려면

이 예제에서는 키가 '프로젝트'이고 값이 'secretProject'인 모든 비용 할당 태그를 검색합니다.

명령:

```
aws ce get-tags --search-string secretProject --time-
period Start=2017-01-01,End=2017-05-18 --tag-key Project
```

출력:

```
{
  "ReturnSize": 2,
  "Tags": [
    "secretProject1",
    "secretProject2"
  ],
  "TotalSize": 2
}
```

- 자세한 API 내용은 명령 참조GetTags의 섹션을 참조하세요. AWS CLI

## 를 사용한 Firehose 예제 AWS CLI

다음 코드 예제에서는 Firehose와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### `list-delivery-streams`

다음 코드 예시에서는 `list-delivery-streams`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 전송 스트림을 나열하려면

다음 `list-delivery-streams` 예제에서는 AWS 계정에서 사용 가능한 전송 스트림을 나열합니다.

```
aws firehose list-delivery-streams
```

출력:

```
{
    "DeliveryStreamNames": [
        "my-stream"
    ],
    "HasMoreDeliveryStreams": false
}
```

자세한 정보는 Amazon Kinesis Data Firehose 개발자 안내서의 Amazon Kinesis Data Firehose 전송 스트림을 참조하세요.

- 자세한 API 내용은 명령 참조ListDeliveryStreams의 섹션을 참조하세요. AWS CLI

### `put-record-batch`

다음 코드 예시에서는 `put-record-batch`을 사용하는 방법을 보여 줍니다.

AWS CLI

스트림에 여러 레코드를 쓰는 방법

다음 put-record-batch 예시에서는 하나의 스트림에 3개의 레코드를 씁니다. 데이터는 Base64 형식으로 인코딩됩니다.

```
aws firehose put-record-batch \
    --delivery-stream-name my-stream \
    --records file://records.json
```

myfile.json의 콘텐츠:

```
[
    {"Data": "Rmlyc3QgdGhpbmc="},
    {"Data": "U2Vjb25kIHRoaW5n"},
    {"Data": "VGhpcmQgdGhpbmc="}
]
```

출력:

```
{
    "FailedPutCount": 0,
    "Encrypted": false,
    "RequestResponses": [
        {
            "RecordId": "9D2OJ6t2EqCTZTXwGzeSv/EVHxRoRCw89xd+o3+sXg8DhYOaWKPSmZy/
CGlRVEys1u1xbeKh6VofEYKkoeiDrcjrxhQp9iF7sUW7pujiMEQ5LzlrzCkGosxQn
+3boDnURDEaD42V7GiixpOyLJkYZcae1i7HzlCEoy9LJhMr8EjDSi4Om/9Vc2uhwwuAtGE0XKpxJ2WD7ZRWtAnYlKAnv
        },
        {
            "RecordId": "jFirejqxCLlK5xjH/UNmlMVcjktEN76I7916X9PaZ
+PVaOSXDfU1WGOqEZhxq2js7xcZ552eoeDxsuTU1MSq9nZTbVfb6cQTIXnm/GsuF37Uhg67GKmR5z90l6XKJ
+/+pDloFv7Hh9a3oUS6wYm3DcNRLTHHAimANp1PhkQvWpvLRfzbuCUkBphR2QVzhP9OiHLbzGwy8/
DfH8sqWEUYASNJKS8GXP5s"
        },
        {
            "RecordId":
 "oy0amQ40o5Y2YV4vxzufdcMOOw6n3EPr3tpPJGoYVNKH4APPVqNcbUgefo1stEFRg4hTLrf2k6eliHu/9+YJ5R3iie
DTBt3qBlmTj7Xq8SKVbO1S7YvMTpWkMKA86f8JfmT8BMKoMb4XZS/sOkQLe+qh0sYKXWl"
        }
    ]
```

```
}
```

자세한 내용은 Amazon Kinesis Data Firehose 개발자 안내서의 Amazon Kinesis Data Firehose 전송 스트림으로 데이터 전송을 참조하세요.

- 자세한 API 내용은 명령 참조PutRecordBatch의 섹션을 참조하세요. AWS CLI

**put-record**

다음 코드 예시에서는 put-record을 사용하는 방법을 보여 줍니다.

AWS CLI

스트림에 레코드를 작성하려면

다음 put-record 예제에서는 스트림에 데이터를 씁니다. 데이터는 Base64 형식으로 인코딩됩니다.

```
aws firehose put-record \
    --delivery-stream-name my-stream \
    --record '{"Data":"SGVsbG8gd29ybGQ="}'
```

출력:

```
{
    "RecordId": "RjB5K/nnoGFHqwTsZlNd/
TTqvjE8V5dsyXZTQn2JXrdpMTOwssyEb6nfC8fwf1whhwnItt4mvrn+gsqeK5jB7QjuLg283+Ps4Sz/
j1Xujv31iDhnPdaLw4BOyM9Amv7PcCuB2079RuM0NhoakbyUymlwY8yt20G8X2420wu1jlFafhci4erAt7QhDEvpwuK8
    "Encrypted": false
}
```

자세한 내용은 Amazon Kinesis Data Firehose 개발자 안내서의 Amazon Kinesis Data Firehose 전송 스트림으로 데이터 전송을 참조하세요.

- 자세한 API 내용은 명령 참조PutRecord의 섹션을 참조하세요. AWS CLI

## 를 사용한 Amazon Data Lifecycle Manager 예제 AWS CLI

다음 코드 예제에서는 Amazon Data Lifecycle Manager와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `create-default-role`

다음 코드 예시에서는 `create-default-role`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon에 필요한 IAM 역할을 생성하려면 DLM

다음 `dlm create-default-role` 예제에서는 스냅샷 관리를 위한 AWS DataLifecycleManagerDefaultRole 기본 역할을 생성합니다.

```
aws dlm create-default-role \
    --resource-type snapshot
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Elastic Compute Cloud 사용 설명서의 Amazon Data Lifecycle Manager에 대한 기본 서비스 역할을 참조하세요.

- 자세한 API 내용은 명령 참조CreateDefaultRole의 섹션을 참조하세요. AWS CLI

### `create-lifecycle-policy`

다음 코드 예시에서는 `create-lifecycle-policy`을 사용하는 방법을 보여 줍니다.

AWS CLI

수명 주기 정책을 생성하려면

다음 create-lifecycle-policy 예제에서는 지정된 시간에 볼륨의 일일 스냅샷을 생성하는 수명 주기 정책을 생성합니다. 지정된 태그가 스냅샷에 추가되고 태그도 볼륨에서 복사되어 스냅샷에 추가됩니다. 새 스냅샷 생성이 지정된 최대 수를 초과하는 경우 가장 오래된 스냅샷이 삭제됩니다.

```
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
```

policyDetails.json의 콘텐츠:

```
{
    "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
            }
```

```
        }
    ]
}
```

출력:

```
{
    "PolicyId": "policy-0123456789abcdef0"
}
```

- 자세한 API 내용은 명령 참조CreateLifecyclePolicy의 섹션을 참조하세요. AWS CLI

### delete-lifecycle-policy

다음 코드 예시에서는 delete-lifecycle-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

수명 주기 정책을 삭제하려면

다음 예제에서는 지정된 수명 주기 정책을 삭제합니다.

```
aws dlm delete-lifecycle-policy --policy-id policy-0123456789abcdef0
```

- 자세한 API 내용은 명령 참조DeleteLifecyclePolicy의 섹션을 참조하세요. AWS CLI

### get-lifecycle-policies

다음 코드 예시에서는 get-lifecycle-policies을 사용하는 방법을 보여 줍니다.

AWS CLI

수명 주기 정책 요약을 가져오려면

다음 get-lifecycle-policies 예제에서는 모든 수명 주기 정책을 나열합니다.

```
aws dlm get-lifecycle-policies
```

출력:

```
{
    "Policies": [
        {
            "PolicyId": "policy-0123456789abcdef0",
            "Description": "My first policy",
            "State": "ENABLED"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetLifecyclePolicies의 섹션을 참조하세요. AWS CLI

### get-lifecycle-policy

다음 코드 예시에서는 get-lifecycle-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

수명 주기 정책을 설명하려면

다음 get-lifecycle-policy 예제에서는 지정된 수명 주기 정책에 대한 세부 정보를 표시합니다.

```
aws dlm get-lifecycle-policy \
    --policy-id policy-0123456789abcdef0
```

출력:

```
{
    "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
                  }
              }
            ]
        }
    }
}
```

- 자세한 API 내용은 명령 참조GetLifecyclePolicy의 섹션을 참조하세요. AWS CLI

### update-lifecycle-policy

다음 코드 예시에서는 update-lifecycle-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 수명 주기 정책을 활성화하려면

다음 update-lifecycle-policy 예제에서는 지정된 수명 주기 정책을 활성화합니다.

```
aws dlm update-lifecycle-policy \
    --policy-id policy-0123456789abcdef0 \
    --state ENABLED
```

예제 2: 수명 주기 정책을 비활성화하려면

다음 update-lifecycle-policy 예제에서는 지정된 수명 주기 정책을 비활성화합니다.

```
aws dlm update-lifecycle-policy \
    --policy-id policy-0123456789abcdef0 \
    --state DISABLED
```

예제 3: 수명 주기 정책의 세부 정보를 업데이트하려면

다음 update-lifecycle-policy 예제에서는 지정된 수명 주기 정책의 대상 태그를 업데이트합니다.

```
aws dlm update-lifecycle-policy \
    --policy-id policy-0123456789abcdef0
    --policy-details file://policyDetails.json
```

policyDetails.json의 콘텐츠. 이 파일에서 참조되지 않은 기타 세부 정보는 명령에 의해 변경되지 않습니다.

```
{
    "TargetTags": [
        {
            "Key": "costCenter",
            "Value": "120"
        },
        {
            "Key": "project",
            "Value": "lima"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조UpdateLifecyclePolicy의 섹션을 참조하세요. AWS CLI

# AWS Data Pipeline 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS Data Pipeline.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `activate-pipeline`

다음 코드 예시에서는 `activate-pipeline`을 사용하는 방법을 보여 줍니다.

AWS CLI

파이프라인을 활성화하려면

이 예제에서는 지정된 파이프라인을 활성화합니다.

```
aws datapipeline activate-pipeline --pipeline-id df-00627471SOVYZEXAMPLE
```

특정 날짜 및 시간에 파이프라인을 활성화하려면 다음 명령을 사용합니다.

```
aws datapipeline activate-pipeline --pipeline-id df-00627471SOVYZEXAMPLE --start-
timestamp 2015-04-07T00:00:00Z
```

- 자세한 API 내용은 명령 참조ActivatePipeline의 섹션을 참조하세요. AWS CLI

### `add-tags`

다음 코드 예시에서는 `add-tags`을 사용하는 방법을 보여 줍니다.

AWS CLI

파이프라인에 태그를 추가하려면

이 예제에서는 지정된 파이프라인에 지정된 태그를 추가합니다.

```
aws datapipeline add-tags --pipeline-id df-00627471SOVYZEXAMPLE --
tags key=environment,value=production key=owner,value=sales
```

태그를 보려면 describe-pipelines 명령을 사용합니다. 예를 들어 예제 명령에 추가된 태그는 describe-pipelines 출력에 다음과 같이 표시됩니다.

```
{
    ...
        "tags": [
            {
                "value": "production",
                "key": "environment"
            },
            {
                "value": "sales",
                "key": "owner"
            }
        ]
    ...
}
```

- 자세한 API 내용은 명령 참조AddTags의 섹션을 참조하세요. AWS CLI

### `create-pipeline`

다음 코드 예시에서는 `create-pipeline`을 사용하는 방법을 보여 줍니다.

AWS CLI

파이프라인을 생성하려면

이 예제에서는 파이프라인을 생성합니다.

```
aws datapipeline create-pipeline --name my-pipeline --unique-id my-pipeline-token
```

다음은 예 출력입니다.

```
{
    "pipelineId": "df-00627471SOVYZEXAMPLE"
}
```

- 자세한 API 내용은 명령 참조CreatePipeline의 섹션을 참조하세요. AWS CLI

### deactivate-pipeline

다음 코드 예시에서는 deactivate-pipeline을 사용하는 방법을 보여 줍니다.

AWS CLI

파이프라인을 비활성화하려면

이 예제에서는 지정된 파이프라인을 비활성화합니다.

```
aws datapipeline deactivate-pipeline --pipeline-id df-00627471SOVYZEXAMPLE
```

실행 중인 모든 활동이 완료된 후에만 파이프라인을 비활성화하려면 다음 명령을 사용합니다.

```
aws datapipeline deactivate-pipeline --pipeline-id df-00627471SOVYZEXAMPLE --no-
cancel-active
```

- 자세한 API 내용은 명령 참조DeactivatePipeline의 섹션을 참조하세요. AWS CLI

### delete-pipeline

다음 코드 예시에서는 delete-pipeline을 사용하는 방법을 보여 줍니다.

AWS CLI

파이프라인을 삭제하려면

이 예제에서는 지정된 파이프라인을 삭제합니다.

```
aws datapipeline delete-pipeline --pipeline-id df-00627471SOVYZEXAMPLE
```

- 자세한 API 내용은 명령 참조DeletePipeline의 섹션을 참조하세요. AWS CLI

### describe-pipelines

다음 코드 예시에서는 describe-pipelines을 사용하는 방법을 보여 줍니다.

AWS CLI

파이프라인을 설명하려면

이 예제에서는 지정된 파이프라인을 설명합니다.

```
aws datapipeline describe-pipelines --pipeline-ids df-00627471SOVYZEXAMPLE
```

다음은 예 출력입니다.

```
{
  "pipelineDescriptionList": [
      {
          "fields": [
              {
                  "stringValue": "PENDING",
                  "key": "@pipelineState"
              },
              {
                  "stringValue": "my-pipeline",
                  "key": "name"
              },
              {
                  "stringValue": "2015-04-07T16:05:58",
                  "key": "@creationTime"
              },
              {
                  "stringValue": "df-00627471SOVYZEXAMPLE",
                  "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-00627471SOVYZEXAMPLE",
            "name": "my-pipeline",
            "tags": []
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribePipelines의 섹션을 참조하세요. AWS CLI

### get-pipeline-definition

다음 코드 예시에서는 get-pipeline-definition을 사용하는 방법을 보여 줍니다.

AWS CLI

파이프라인 정의를 가져오려면

이 예제에서는 지정된 파이프라인에 대한 파이프라인 정의를 가져옵니다.

```
aws datapipeline get-pipeline-definition --pipeline-id df-00627471SOVYZEXAMPLE
```

다음은 예 출력입니다.

```
{
  "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"
  }
}
```

- 자세한 API 내용은 명령 참조GetPipelineDefinition의 섹션을 참조하세요. AWS CLI

### `list-pipelines`

다음 코드 예시에서는 `list-pipelines`을 사용하는 방법을 보여 줍니다.

AWS CLI

파이프라인을 나열하려면

이 예제에서는 파이프라인을 나열합니다.

```
aws datapipeline list-pipelines
```

다음은 예 출력입니다.

```
{
  "pipelineIdList": [
      {
          "id": "df-00627471SOVYZEXAMPLE",
          "name": "my-pipeline"
      },
      {
          "id": "df-09028963KNVMREXAMPLE",
          "name": "ImportDDB"
      },
      {
          "id": "df-0870198233ZYVEXAMPLE",
          "name": "CrossRegionDDB"
      },
      {
          "id": "df-00189603TB4MZEXAMPLE",
          "name": "CopyRedshift"
      }
  ]
}
```

- 자세한 API 내용은 명령 참조ListPipelines의 섹션을 참조하세요. AWS CLI

### `list-runs`

다음 코드 예시에서는 `list-runs`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 파이프라인 실행을 나열하려면

다음 `list-runs` 예제에서는 지정된 파이프라인에 대한 실행을 나열합니다.

```
aws datapipeline list-runs --pipeline-id df-00627471SOVYZEXAMPLE
```

출력:

```
    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
```

예제 2: 지정된 날짜 사이의 파이프라인 실행을 나열하려면

다음 `list-runs` 예제에서는 `--start-interval`를 사용하여 출력에 포함할 날짜를 지정합니다.

```
aws datapipeline list-runs --pipeline-id df-01434553B58A2SHZUKO5 --start-
interval 2017-10-07T00:00:00,2017-10-08T00:00:00
```

- 자세한 API 내용은 명령 참조ListRuns의 섹션을 참조하세요. AWS CLI

### put-pipeline-definition

다음 코드 예시에서는 `put-pipeline-definition`을 사용하는 방법을 보여 줍니다.

AWS CLI

파이프라인 정의를 업로드하려면

이 예제에서는 지정된 파이프라인 정의를 지정된 파이프라인에 업로드합니다.

```
aws datapipeline put-pipeline-definition --pipeline-id df-00627471SOVYZEXAMPLE --
pipeline-definition file://my-pipeline-definition.json
```

다음은 예 출력입니다.

```
{
  "validationErrors": [],
  "errored": false,
```

```
  "validationWarnings": []
}
```

- 자세한 API 내용은 명령 참조PutPipelineDefinition의 섹션을 참조하세요. AWS CLI

### `remove-tags`

다음 코드 예시에서는 `remove-tags`을 사용하는 방법을 보여 줍니다.

AWS CLI

파이프라인에서 태그를 제거하려면

이 예제에서는 지정된 파이프라인에서 지정된 태그를 제거합니다.

```
aws datapipeline remove-tags --pipeline-id df-00627471SOVYZEXAMPLE --tag-
keys environment
```

- 자세한 API 내용은 명령 참조RemoveTags의 섹션을 참조하세요. AWS CLI

## DataSync 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 DataSync.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

### `update-location-azure-blob`

다음 코드 예시에서는 `update-location-azure-blob`을 사용하는 방법을 보여 줍니다.

AWS CLI

새 에이전트를 사용하여 전송 위치를 업데이트하려면

다음 update-location-object-storage 예제에서는 새 에이전트를 사용하여 Microsoft Azure Blob Storage의 DataSync 위치를 업데이트합니다.

```
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" \
    }'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS DataSync 사용 설명서의 에이전트 교체를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateLocationAzureBlob의 섹션을 참조하세요. AWS CLI

### update-location-hdfs

다음 코드 예시에서는 update-location-hdfs을 사용하는 방법을 보여 줍니다.

AWS CLI

새 에이전트를 사용하여 전송 위치를 업데이트하려면

다음 update-location-hdfs 예제에서는 새 에이전트로 위치를 업데이트 DataSync HDFS 합니다. HDFS 클러스터가 Kerberos 인증을 사용하는 경우에만 --kerberos-keytab 및 --kerberos-krb5-conf 옵션이 필요합니다.

```
aws datasync update-location-hdfs \
    --location-arn arn:aws:datasync:us-west-2:123456789012:location/loc-
abcdef01234567890 \
    --agent-arns arn:aws:datasync:us-west-2:123456789012:agent/
agent-1234567890abcdef0 \
    --kerberos-keytab file://hdfs.keytab
    --kerberos-krb5-conf file://krb5.conf
```

hdfs.keytab의 콘텐츠:

```
N/A. The content of this file is encrypted and not human readable.
```

krb5.conf의 콘텐츠:

```
[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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS DataSync 사용 설명서의 에이전트 교체를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateLocationHdfs의 섹션을 참조하세요. AWS CLI

### update-location-nfs

다음 코드 예시에서는 update-location-nfs을 사용하는 방법을 보여 줍니다.

AWS CLI

새 에이전트를 사용하여 전송 위치를 업데이트하려면

다음 update-location-nfs 예제에서는 새 에이전트로 위치를 업데이트 DataSync NFS합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS DataSync 사용 설명서의 에이전트 교체를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateLocationNfs의 섹션을 참조하세요. AWS CLI

### update-location-object-storage

다음 코드 예시에서는 update-location-object-storage을 사용하는 방법을 보여 줍니다.

AWS CLI

새 에이전트를 사용하여 전송 위치를 업데이트하려면

다음 update-location-object-storage 예제에서는 새 에이전트를 사용하여 DataSync 객체 스토리지 위치를 업데이트합니다.

```
aws datasync update-location-object-storage \
    --location-arn arn:aws:datasync:us-west-2:123456789012:location/loc-
abcdef01234567890 \
    --agent-arns arn:aws:datasync:us-west-2:123456789012:agent/
agent-1234567890abcdef0 \
    --secret-key secret-key-for-object-storage
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS DataSync 사용 설명서의 에이전트 교체를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateLocationObjectStorage의 섹션을 참조하세요. AWS CLI

**update-location-smb**

다음 코드 예시에서는 update-location-smb을 사용하는 방법을 보여 줍니다.

AWS CLI

새 에이전트를 사용하여 전송 위치를 업데이트하려면

다음 update-location-smb 예제에서는 새 에이전트로 위치를 DataSync SMB 업데이트합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS DataSync 사용 설명서의 에이전트 교체를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateLocationSmb의 섹션을 참조하세요. AWS CLI

## DAX 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다DAX.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### create-cluster

다음 코드 예시에서는 create-cluster을 사용하는 방법을 보여 줍니다.

AWS CLI

DAX 클러스터를 생성하려면

다음 create-cluster 예제에서는 지정된 설정을 사용하여 DAX 클러스터를 생성합니다.

```
aws dax create-cluster \
    --cluster-name daxcluster \
    --node-type dax.r4.large \
    --replication-factor 3 \
    --iam-role-arn roleARN  \
    --sse-specification Enabled=true
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 3단계: DAX 클러스터 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateCluster의 섹션을 참조하세요. AWS CLI

### `create-parameter-group`

다음 코드 예시에서는 `create-parameter-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

파라미터 그룹을 생성하려면

다음 `` create-parameter-group`` 예제에서는 지정된 설정을 사용하여 파라미터 그룹을 생성합니다.

```
aws dax create-parameter-group \
    --parameter-group-name daxparametergroup \
    --description "A new parameter group"
```

출력:

```
{
    "ParameterGroup": {
        "ParameterGroupName": "daxparametergroup",
        "Description": "A new parameter group"
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DAX 클러스터 관리를 참조하세요.

- 자세한 API 내용은 명령 참조CreateParameterGroup의 섹션을 참조하세요. AWS CLI

### `create-subnet-group`

다음 코드 예시에서는 `create-subnet-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

DAX 서브넷 그룹을 생성하려면

다음 `create-subnet-group` 예제에서는 지정된 설정을 사용하여 서브넷 그룹을 생성합니다.

```
aws dax create-subnet-group \
    --subnet-group-name daxSubnetGroup \
    --subnet-ids subnet-11111111 subnet-22222222
```

출력:

```
{
    "SubnetGroup": {
        "SubnetGroupName": "daxSubnetGroup",
        "VpcId": "vpc-05a1fa8e00c325226",
        "Subnets": [
            {
                "SubnetIdentifier": "subnet-11111111",
                "SubnetAvailabilityZone": "us-west-2b"
            },
            {
                "SubnetIdentifier": "subnet-22222222",
                "SubnetAvailabilityZone": "us-west-2c"
            }
        ]
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 2단계: 서브넷 그룹 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateSubnetGroup의 섹션을 참조하세요. AWS CLI

### `decrease-replication-factor`

다음 코드 예시에서는 `decrease-replication-factor`을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터에서 하나 이상의 노드를 제거하려면

다음 `decrease-replication-factor` 예제에서는 지정된 DAX 클러스터의 노드 수를 1로 줄입니다.

```
aws dax decrease-replication-factor \
    --cluster-name daxcluster \
    --new-replication-factor 1
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DAX 클러스터 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DecreaseReplicationFactor의 섹션을 참조하세요. AWS CLI

### delete-cluster

다음 코드 예시에서는 delete-cluster을 사용하는 방법을 보여 줍니다.

AWS CLI

DAX 클러스터를 삭제하려면

다음 delete-cluster 예제에서는 지정된 DAX 클러스터를 삭제합니다.

```
aws dax delete-cluster \
    --cluster-name daxcluster
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DAX 클러스터 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteCluster의 섹션을 참조하세요. AWS CLI

### delete-parameter-group

다음 코드 예시에서는 delete-parameter-group을 사용하는 방법을 보여 줍니다.

AWS CLI

파라미터 그룹을 삭제하려면

다음 delete-parameter-group 예제에서는 지정된 DAX 파라미터 그룹을 삭제합니다.

```
aws dax  delete-parameter-group \
    --parameter-group-name daxparametergroup
```

출력:

```
{
    "DeletionMessage": "Parameter group daxparametergroup has been deleted."
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DAX 클러스터 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteParameterGroup의 섹션을 참조하세요. AWS CLI

### delete-subnet-group

다음 코드 예시에서는 delete-subnet-group을 사용하는 방법을 보여 줍니다.

AWS CLI

서브넷 그룹을 삭제하려면

다음 delete-subnet-group 예제에서는 지정된 DAX 서브넷 그룹을 삭제합니다.

```
aws dax delete-subnet-group \
    --subnet-group-name daxSubnetGroup
```

출력:

```
{
    "DeletionMessage": "Subnet group daxSubnetGroup has been deleted."
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DAX 클러스터 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteSubnetGroup의 섹션을 참조하세요. AWS CLI

### describe-clusters

다음 코드 예시에서는 describe-clusters을 사용하는 방법을 보여 줍니다.

AWS CLI

프로비저닝된 모든 DAX 클러스터에 대한 정보를 반환하려면

다음 describe-clusters 예제에서는 프로비저닝된 모든 DAX 클러스터에 대한 세부 정보를 표시합니다.

```
aws dax describe-clusters
```

출력:

```
{
    "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"
            }
        }
    ]
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DAX 클러스터 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeClusters의 섹션을 참조하세요. AWS CLI

### describe-default-parameters

다음 코드 예시에서는 describe-default-parameters을 사용하는 방법을 보여 줍니다.

AWS CLI

에 대한 기본 시스템 파라미터 정보를 반환하려면 DAX

다음 `describe-default-parameters` 예제에서는 에 대한 기본 시스템 파라미터 정보를 표시합니다DAX.

```
aws dax describe-default-parameters
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DAX 클러스터 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeDefaultParameters의 섹션을 참조하세요. AWS CLI

### `describe-events`

다음 코드 예시에서는 `describe-events`을 사용하는 방법을 보여 줍니다.

AWS CLI

DAX 클러스터 및 파라미터 그룹과 관련된 모든 이벤트를 반환하려면

다음 `describe-events` 예제에서는 DAX 클러스터 및 파라미터 그룹과 관련된 이벤트의 세부 정보를 보여줍니다.

```
aws dax describe-events
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DAX 클러스터 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeEvents의 섹션을 참조하세요. AWS CLI

### `describe-parameter-groups`

다음 코드 예시에서는 `describe-parameter-groups`을 사용하는 방법을 보여 줍니다.

AWS CLI

에 정의된 파라미터 그룹을 설명하려면 DAX

다음 `describe-parameter-groups` 예제에서는 에 정의된 파라미터 그룹에 대한 세부 정보를 검색합니다DAX.

```
aws dax describe-parameter-groups
```

출력:

```
{
    "ParameterGroups": [
        {
            "ParameterGroupName": "default.dax1.0",
            "Description": "Default parameter group for dax1.0"
        }
    ]
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DAX 클러스터 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeParameterGroups의 섹션을 참조하세요. AWS CLI

### describe-parameters

다음 코드 예시에서는 describe-parameters을 사용하는 방법을 보여 줍니다.

AWS CLI

파라미터 그룹에 정의된 DAX 파라미터를 설명하려면

다음 describe-parameters 예제에서는 지정된 파라미터 그룹에 정의된 DAX 파라미터에 대한 세부 정보를 검색합니다.

```
aws dax describe-parameters \
    --parameter-group-name default.dax1.0
```

출력:

```
{
    "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"
        }
```

```
    ]
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DAX 클러스터 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeParameters의 섹션을 참조하세요. AWS CLI

### `describe-subnet-groups`

다음 코드 예시에서는 `describe-subnet-groups`을 사용하는 방법을 보여 줍니다.

AWS CLI

에 정의된 서브넷 그룹을 설명하려면 DAX

다음 `describe-subnet-groups` 예제에서는 에 정의된 서브넷 그룹에 대한 세부 정보를 검색합니다DAX.

```
aws dax describe-subnet-groups
```

출력:

```
{
    "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"
                }
            ]
        }
    ]
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DAX 클러스터 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeSubnetGroups의 섹션을 참조하세요. AWS CLI

### `increase-replication-factor`

다음 코드 예시에서는 `increase-replication-factor`을 사용하는 방법을 보여 줍니다.

AWS CLI

DAX 클러스터의 복제 인수를 늘리려면

다음 `increase-replication-factor` 예제에서는 지정된 DAX 클러스터의 복제 인수를 3으로 늘립니다.

```
aws dax increase-replication-factor \
    --cluster-name daxcluster \
    --new-replication-factor 3
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DAX 클러스터 관리를 참조하세요.

- 자세한 API 내용은 명령 참조IncreaseReplicationFactor의 섹션을 참조하세요. AWS CLI

### list-tags

다음 코드 예시에서는 list-tags을 사용하는 방법을 보여 줍니다.

AWS CLI

DAX 리소스의 태그를 나열하려면

다음 list-tags 예제에서는 지정된 DAX 클러스터에 연결된 태그 키와 값을 나열합니다.

```
aws dax list-tags \
    --resource-name arn:aws:dax:us-west-2:123456789012:cache/daxcluster
```

출력:

```
{
    "Tags": [
        {
            "Key": "ClusterUsage",
            "Value": "prod"
        }
    ]
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DAX 클러스터 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ListTags의 섹션을 참조하세요. AWS CLI

### tag-resource

다음 코드 예시에서는 tag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

DAX 리소스에 태그를 지정하려면

다음 tag-resource 예제에서는 지정된 태그 키 이름과 관련 값을 지정된 DAX 클러스터에 연결하여 클러스터 사용량을 설명합니다.

```
aws dax tag-resource \
    --resource-name arn:aws:dax:us-west-2:123456789012:cache/daxcluster \
```

```
    --tags="Key=ClusterUsage,Value=prod"
```

출력:

```
{
    "Tags": [
        {
            "Key": "ClusterUsage",
            "Value": "prod"
        }
    ]
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DAX 클러스터 관리를 참조하세요.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### `untag-resource`

다음 코드 예시에서는 `untag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

DAX 리소스에서 태그를 제거하려면

다음 `untag-resource` 예제에서는 DAX 클러스터에서 지정된 키 이름이 있는 태그를 제거합니다.

```
aws dax untag-resource  \
    --resource-name arn:aws:dax:us-west-2:123456789012:cache/daxcluster \
    --tag-keys="ClusterUsage"
```

출력:

```
{
    "Tags": []
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DAX 클러스터 관리를 참조하세요.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

# 를 사용한 Detective 예제 AWS CLI

다음 코드 예제에서는 Detective AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `accept-invitation`

다음 코드 예시에서는 `accept-invitation`을 사용하는 방법을 보여 줍니다.

AWS CLI

동작 그래프에서 멤버 계정이 되기 위한 초대를 수락하려면

다음 `accept-invitation` 예제에서는 동작 그래프 arn:aws:detective:us-east-1:111122223333:graph:123412341234에서 멤버 계정이 되기 위한 초대를 수락합니다.

```
aws detective accept-invitation \
    --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Detective 관리 안내서의 동작 그래프 초대에 대한 응답을 참조하세요.

- 자세한 API 내용은 명령 참조AcceptInvitation의 섹션을 참조하세요. AWS CLI

### `create-graph`

다음 코드 예시에서는 `create-graph`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon Detective를 활성화하고 새 동작 그래프를 생성하려면

다음 create-graph 예제에서는 명령이 실행되는 리전에서 명령을 실행하는 AWS 계정에 대해 Detective를 활성화합니다. 해당 계정이 관리자 계정인 새 동작 그래프가 생성됩니다. 명령은 또한 Finance 값을 Department 태그에 할당합니다.

```
aws detective create-graph \
    --tags '{"Department": "Finance"}'
```

출력:

```
{
    "GraphArn": "arn:aws:detective:us-
east-1:111122223333:graph:027c7c4610ea4aacaf0b883093cab899"
}
```

자세한 내용은 Amazon Detective 관리 안내서의 Amazon Detective 활성화를 참조하세요.

- 자세한 API 내용은 명령 참조CreateGraph의 섹션을 참조하세요. AWS CLI

## create-members

다음 코드 예시에서는 create-members을 사용하는 방법을 보여 줍니다.

AWS CLI

멤버 계정을 동작 그래프에 초대하려면

다음 create-members 예제에서는 동작 그래프 arn:aws:detective:us-east-1:111122223333:graph:123412341234에서 두 AWS 계정을 멤버 계정으로 초대합니다. 각 계정에 대해 요청은 AWS 계정 ID와 계정 루트 사용자 이메일 주소를 제공합니다. 요청에 초대 이메일에 삽입할 사용자 지정 메시지가 포함되어 있습니다.

```
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."
```

출력:

```
{
    "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": [ ]
}
```

자세한 내용은 Amazon Detective 관리 안내서의 동작 그래프<https://docs.aws.amazon.com/detective/latest/adminguide/graph-admin-add-member-accounts.html>에 멤버 계정 초대를 참조하세요.

초대 이메일을 보내지 않고 멤버 계정을 초대하려면

다음 `create-members` 예제에서는 동작 그래프 arn:aws:detective:us-east-1:111122223333:graph:123412341234에서 두 AWS 계정을 멤버 계정으로 초대합니다. 각 계정에 대해 요청은 AWS 계정 ID와 계정 루트 사용자 이메일 주소를 제공합니다. 멤버 계정은 초대 이메일을 받지 않습니다.

```
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
```

출력:

```
{
    "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": [ ]
}
```

자세한 내용은 Amazon Detective 관리 안내서의 동작 그래프<https://docs.aws.amazon.com/detective/latest/adminguide/graph-admin-add-member-accounts.html>에 멤버 계정 초대를 참조하세요.

- 자세한 API 내용은 명령 참조CreateMembers의 섹션을 참조하세요. AWS CLI

### delete-graph

다음 코드 예시에서는 delete-graph을 사용하는 방법을 보여 줍니다.

AWS CLI

Detective를 비활성화하고 동작 그래프를 삭제하려면

다음 delete-graph 예제에서는 Detective를 비활성화하고 지정된 동작 그래프를 삭제합니다.

```
aws detective delete-graph \
    --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Detective 관리 안내서의 Amazon Detective 비활성화를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteGraph의 섹션을 참조하세요. AWS CLI

### delete-members

다음 코드 예시에서는 delete-members을 사용하는 방법을 보여 줍니다.

AWS CLI

동작 그래프에서 멤버 계정을 제거하려면

다음 delete-members 예제에서는 동작 그래프 arn:aws:detective:us-east-1:111122223333:graph:123412341234에서 두 멤버 계정을 제거합니다. 계정을 식별하기 위해 요청은 AWS 계정 를 제공합니다IDs.

```
aws detective delete-members \
    --account-ids 444455556666 123456789012 \
    --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

출력:

```
 {
    "AccountIds": [ "444455556666", "123456789012" ],
    "UnprocessedAccounts": [ ]
}
```

자세한 내용은 Amazon Detective 관리 안내서의 동작 그래프<https://docs.aws.amazon.com/detective/latest/adminguide/graph-admin-remove-member-accounts.html>에서 멤버 계정 제거를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteMembers의 섹션을 참조하세요. AWS CLI

### `disassociate-membership`

다음 코드 예시에서는 `disassociate-membership`을 사용하는 방법을 보여 줍니다.

AWS CLI

동작 그래프에서 멤버십을 사직하려면

다음 연결 해제-멤버십 예제는 동작 그래프 arn:aws:detective:us-east-1:111122223333:graph:123412341234에서 명령을 실행하는 AWS 계정을 제거합니다.

```
aws detective disassociate-membership \
     --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

자세한 내용은 Amazon Detective 관리 안내서의 동작 그래프에서 계정 제거<https://docs.aws.amazon.com/detective/latest/adminguide/member-remove-self-from-graph.html>를 참조하세요.

- 자세한 API 내용은 명령 참조DisassociateMembership의 섹션을 참조하세요. AWS CLI

### `get-members`

다음 코드 예시에서는 `get-members`을 사용하는 방법을 보여 줍니다.

AWS CLI

선택한 동작 그래프 멤버 계정에 대한 정보를 검색하려면

다음 `get-members` 예제에서는 동작 그래프 arn:aws:detective:us-east-1:111122223333:graph:123412341234에서 두 멤버 계정에 대한 정보를 검색합니다. 두 계정의 경우 요청은 AWS 계정 를 제공합니다IDs.

```
aws detective get-members \
    --account-ids 444455556666 123456789012 \
    --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

출력:

```
{
    "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": [ ]
}
```

자세한 내용은 Amazon Detective 관리 안내서의 동작 그래프<https://docs.aws.amazon.com/detective/latest/adminguide/graph-admin-view-accounts.html>에서 계정 목록 보기를 참조하세요.

- 자세한 API 내용은 명령 참조GetMembers의 섹션을 참조하세요. AWS CLI

### `list-graphs`

다음 코드 예시에서는 `list-graphs`을 사용하는 방법을 보여 줍니다.

AWS CLI

계정이 관리자인 동작 그래프 목록을 보려면

다음 `list-graphs` 예제에서는 현재 리전 내에서 호출 계정이 관리자인 동작 그래프를 검색합니다.

```
aws detective list-graphs
```

출력:

```
{
    "GraphList": [
        {
            "Arn": "arn:aws:detective:us-east-1:111122223333:graph:123412341234",
            "CreatedTime": 1579736111000
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListGraphs의 섹션을 참조하세요. AWS CLI

### `list-invitations`

다음 코드 예시에서는 `list-invitations`을 사용하는 방법을 보여 줍니다.

AWS CLI

계정이 멤버이거나 초대된 동작 그래프 목록을 보려면

다음 `list-invitations` 예제에서는 호출 계정이 초대된 동작 그래프를 검색합니다. 결과에는 열린 초대와 수락된 초대만 포함됩니다. 거부된 초대 또는 제거된 멤버십은 포함되지 않습니다.

```
aws detective list-invitations
```

출력:

```
{
    "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
```

```
    }
]
}
```

자세한 내용은 Amazon Detective 관리 안내서의 동작 그래프 초대 목록 보기<https://docs.aws.amazon.com/detective/latest/adminguide/member-view-graph-invitations.html>를 참조하세요.

- 자세한 API 내용은 명령 참조ListInvitations의 섹션을 참조하세요. AWS CLI

### `list-members`

다음 코드 예시에서는 `list-members`을 사용하는 방법을 보여 줍니다.

AWS CLI

동작 그래프에 멤버 계정을 나열하려면

다음 `list-members` 예제에서는 동작 그래프 에 대해 초대되고 활성화된 멤버 계정을 검색합니다`arn:aws:detective:us-east-1:111122223333:graph:123412341234`. 결과에는 제거된 멤버 계정이 포함되지 않습니다.

```
aws detective list-members \
    --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 Amazon Detective 관리 안내서의 동작 그래프에서 계정 목록 보기를 참조하세요.

- 자세한 API 내용은 명령 참조ListMembers의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

동작 그래프에 할당된 태그를 검색하려면

다음 `list-tags-for-resource` 예제에서는 지정된 동작 그래프에 할당된 태그를 반환합니다.

```
aws detective list-tags-for-resource \
    --resource-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

출력:

```
{
    "Tags": {
        "Department" : "Finance"
    }
}
```

자세한 내용은 Amazon Detective 관리 안내서의 동작 그래프에 대한 태그 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### `reject-invitation`

다음 코드 예시에서는 `reject-invitation`을 사용하는 방법을 보여 줍니다.

AWS CLI

동작 그래프에서 멤버 계정이 되기 위한 초대를 거부하려면

다음 `reject-invitation` 예제에서는 동작 그래프 arn:aws:detective:us-east-1:111122223333:graph:123412341234에서 멤버 계정이 되기 위한 초대를 거부합니다.

```
aws detective reject-invitation \
    --graph-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Detective 관리 안내서의 동작 그래프 초대에 응답<https://docs.aws.amazon.com/detective/latest/adminguide/member-invitation-response.html>을 참조하세요.

- 자세한 API 내용은 명령 참조RejectInvitation의 섹션을 참조하세요. AWS CLI

### `tag-resource`

다음 코드 예시에서는 `tag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 태그를 할당하려면

다음 `tag-resource` 예제에서는 부서 태그의 값을 지정된 동작 그래프에 할당합니다.

```
aws detective tag-resource \
    --resource-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234 \
    --tags '{"Department":"Finance"}'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Detective 관리 안내서의 동작 그래프에 대한 태그 관리를 참조하세요.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

**untag-resource**

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에서 태그 값을 제거하려면

다음 untag-resource 예제에서는 지정된 동작 그래프에서 부서 태그를 제거합니다.

```
aws detective untag-resource \
    --resource-arn arn:aws:detective:us-east-1:111122223333:graph:123412341234 \
    --tag-keys "Department"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Detective 관리 안내서의 동작 그래프에 대한 태그 관리를 참조하세요.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

## 를 사용한 Device Farm 예제 AWS CLI

다음 코드 예제에서는 Device Farm과 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

**create-device-pool**

다음 코드 예시에서는 create-device-pool을 사용하는 방법을 보여 줍니다.

AWS CLI

디바이스 풀을 생성하려면

다음 명령은 프로젝트에 대한 Android 디바이스 풀을 생성합니다.

```
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"
```

`create-project` 또는 출력ARN에서 프로젝트를 가져올 수 있습니다`list-projects`. 파일은 디바이스 플랫폼을 지정하는 현재 폴더의 JSON 문서`device-pool-rules.json`입니다.

```
[
    {
        "attribute": "PLATFORM",
        "operator": "EQUALS",
        "value": "\"ANDROID\""
    }
]
```

출력:

```
{
    "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"
    }
}
```

- 자세한 API 내용은 명령 참조CreateDevicePool의 섹션을 참조하세요. AWS CLI

### create-project

다음 코드 예시에서는 create-project을 사용하는 방법을 보여 줍니다.

AWS CLI

프로젝트를 생성하려면

다음 명령은 라는 새 프로젝트를 생성합니다my-project.

```
aws devicefarm create-project --name my-project
```

출력:

```
{
    "project": {
        "name": "myproject",
        "arn": "arn:aws:devicefarm:us-
west-2:123456789012:project:070fc3ca-7ec1-4741-9c1f-d3e044efc506",
        "created": 1503612890.057
    }
}
```

- 자세한 API 내용은 명령 참조CreateProject의 섹션을 참조하세요. AWS CLI

### create-upload

다음 코드 예시에서는 create-upload을 사용하는 방법을 보여 줍니다.

AWS CLI

업로드를 생성하려면

다음 명령은 Android 앱에 대한 업로드를 생성합니다.

```
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
```

create-project 또는 list-project의 출력ARN에서 프로젝트를 가져올 수 있습니다.

출력:

```
{
    "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%2Fs3%2Faws4_request&X-Amz-
Signature=05050370c38894ef5bd09f5d009f36fc8f96fa4bb04e1bba9aca71b8dbe49a0f",
        "type": "ANDROID_APP",
        "arn": "arn:aws:devicefarm:us-
west-2:123456789012:upload:070fc3ca-7ec1-4741-9c1f-d3e044efc506/dd72723a-
ae9e-4087-09e6-f4cea3599514"
    }
}
```

출력URL에서 서명된 를 사용하여 Device Farm 에 파일을 업로드합니다.

```
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"
```

- 자세한 API 내용은 명령 참조CreateUpload의 섹션을 참조하세요. AWS CLI

### get-upload

다음 코드 예시에서는 get-upload을 사용하는 방법을 보여 줍니다.

AWS CLI

업로드를 보려면

다음 명령은 업로드에 대한 정보를 검색합니다.

```
aws devicefarm get-upload --arn "arn:aws:devicefarm:us-
west-2:123456789012:upload:070fc3ca-7ec1-4741-9c1f-d3e044efc506/dd72723a-
ae9e-4087-09e6-f4cea3599514"
```

의 출력ARN에서 업로드를 가져올 수 있습니다`create-upload`.

출력:

```
{
    "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\"}"
    }
}
```

- 자세한 API 내용은 명령 참조GetUpload의 섹션을 참조하세요. AWS CLI

### `list-projects`

다음 코드 예시에서는 `list-projects`을 사용하는 방법을 보여 줍니다.

AWS CLI

프로젝트를 나열하려면

다음은 프로젝트 목록을 검색합니다.

```
aws devicefarm list-projects
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListProjects의 섹션을 참조하세요. AWS CLI

## AWS Direct Connect 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS Direct Connect.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `accept-direct-connect-gateway-association-proposal`

다음 코드 예시에서는 `accept-direct-connect-gateway-association-proposal`을 사용하는 방법을 보여 줍니다.

AWS CLI

게이트웨이 연결 제안을 수락하려면

다음은 지정된 제안을 accept-direct-connect-gateway-association-proposal 수락합니다.

```
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"
            }
        ]

  }
}
```

자세한 내용은 AWS Direct Connect 사용 설명서의 Transit Gateway 연결 제안 수락 또는 거부를 참조하세요.

- 자세한 API 내용은 명령 참조AcceptDirectConnectGatewayAssociationProposal의 섹션을 참조하세요. AWS CLI

### allocate-connection-on-interconnect

다음 코드 예시에서는 allocate-connection-on-interconnect을 사용하는 방법을 보여 줍니다.

AWS CLI

상호 연결에서 호스팅 연결을 생성하려면

다음 allocate-connection-on-interconnect 명령은 상호 연결에 호스팅 연결을 생성합니다.

```
aws directconnect allocate-connection-on-interconnect --bandwidth 500Mbps --
connection-name mydcinterconnect --owner-account 123456789012 --interconnect-
id dxcon-fgktov66 --vlan 101
```

출력:

```
{
    "partnerName": "TIVIT",
    "vlan": 101,
    "ownerAccount": "123456789012",
    "connectionId": "dxcon-ffzc51m1",
    "connectionState": "ordering",
    "bandwidth": "500Mbps",
    "location": "TIVIT",
    "connectionName": "mydcinterconnect",
    "region": "sa-east-1"
}
```

- 자세한 API 내용은 명령 참조AllocateConnectionOnInterconnect의 섹션을 참조하세요. AWS CLI

### allocate-hosted-connection

다음 코드 예시에서는 allocate-hosted-connection을 사용하는 방법을 보여 줍니다.

AWS CLI

상호 연결에서 호스팅 연결을 생성하려면

다음 allocate-hosted-connection 예제에서는 지정된 인터커넥트에 호스팅 연결을 생성합니다.

```
aws directconnect allocate-hosted-connection \
    --bandwidth 500Mbps \
    --connection-name mydcinterconnect \
    --owner-account 123456789012
```

```
    -connection-id dxcon-fgktov66
    -vlan 101
```

출력:

```
{
    "partnerName": "TIVIT",
    "vlan": 101,
    "ownerAccount": "123456789012",
    "connectionId": "dxcon-ffzc51m1",
    "connectionState": "ordering",
    "bandwidth": "500Mbps",
    "location": "TIVIT",
    "connectionName": "mydcinterconnect",
    "region": "sa-east-1"
}
```

- 자세한 API 내용은 명령 참조AllocateHostedConnection의 섹션을 참조하세요. AWS CLI

### `allocate-private-virtual-interface`

다음 코드 예시에서는 `allocate-private-virtual-interface`을 사용하는 방법을 보여 줍니다.

AWS CLI

프라이빗 가상 인터페이스를 프로비저닝하려면

다음 `allocate-private-virtual-interface` 명령은 다른 고객이 소유할 프라이빗 가상 인터페이스를 프로비저닝합니다.

```
aws directconnect allocate-private-virtual-interface --connection-id dxcon-
ffjrkx17 --owner-account 123456789012 --new-private-virtual-interface-
allocation virtualInterfaceName=PrivateVirtualInterface,vlan=1000,asn=65000,authKey=asdf34ex
```

출력:

```
{
    "virtualInterfaceState": "confirming",
    "asn": 65000,
    "vlan": 1000,
    "customerAddress": "192.168.1.2/30",
    "ownerAccount": "123456789012",
```

```
    "connectionId": "dxcon-ffjrkx17",
    "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"
}
```

- 자세한 API 내용은 명령 참조AllocatePrivateVirtualInterface의 섹션을 참조하세요. AWS CLI

### `allocate-public-virtual-interface`

다음 코드 예시에서는 `allocate-public-virtual-interface`을 사용하는 방법을 보여 줍니다.

AWS CLI

퍼블릭 가상 인터페이스를 프로비저닝하려면

다음 `allocate-public-virtual-interface` 명령은 다른 고객이 소유할 퍼블릭 가상 인터페이스를 프로비저닝합니다.

```
aws directconnect allocate-public-virtual-interface --connection-id dxcon-
ffjrkx17 --owner-account 123456789012 --new-public-virtual-interface-
allocation virtualInterfaceName=PublicVirtualInterface,vlan=2000,asn=65000,authKey=asdf34exa
{cidr=203.0.113.4/30}]
```

출력:

```
{
    "virtualInterfaceState": "confirming",
    "asn": 65000,
    "vlan": 2000,
    "customerAddress": "203.0.113.2/30",
    "ownerAccount": "123456789012",
```

```
    "connectionId": "dxcon-ffjrkx17",
    "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"
}
```

- 자세한 API 내용은 명령 참조AllocatePublicVirtualInterface의 섹션을 참조하세요. AWS CLI

### `allocate-transit-virtual-interface`

다음 코드 예시에서는 `allocate-transit-virtual-interface`을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 AWS 계정에서 소유할 전송 가상 인터페이스를 프로비저닝하려면

다음 `allocate-transit-virtual-interface` 예제에서는 지정된 계정에 대한 전송 가상 인터페이스를 프로비저닝합니다.

```
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
```

출력:


```
{
    "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>0xzxgA9YoW9h58u8SEXAMPLE</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": "0xzxgA9YoW9h58u8EXAMPLE",
                "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"
            }
        ]
    }
}
```

자세한 내용은 AWS Direct Connect 사용 설명서의 호스팅 전송 가상 인터페이스 생성을 참조하세요.

- 자세한 API 내용은 명령 참조AllocateTransitVirtualInterface의 섹션을 참조하세요. AWS CLI

### `associate-connection-with-lag`

다음 코드 예시에서는 `associate-connection-with-lag`을 사용하는 방법을 보여 줍니다.

AWS CLI

연결을 에 연결하려면 LAG

다음 예제에서는 지정된 연결을 지정된 와 연결합니다LAG.

명령:

```
aws directconnect associate-connection-with-lag --lag-id dxlag-fhccu14t --
connection-id dxcon-fg9607vm
```

출력:

```
{
  "ownerAccount": "123456789012",
  "connectionId": "dxcon-fg9607vm",
  "lagId": "dxlag-fhccu14t",
  "connectionState": "requested",
  "bandwidth": "1Gbps",
  "location": "EqDC2",
  "connectionName": "Con2ForLag",
  "region": "us-east-1"
}
```

- 자세한 API 내용은 명령 참조AssociateConnectionWithLag의 섹션을 참조하세요. AWS CLI

### associate-hosted-connection

다음 코드 예시에서는 associate-hosted-connection을 사용하는 방법을 보여 줍니다.

AWS CLI

호스팅 연결을 에 연결하려면 LAG

다음 예제에서는 지정된 호스팅 연결을 지정된 와 연결합니다LAG.

명령:

```
aws directconnect associate-hosted-connection --parent-connection-id dxlag-fhccu14t
 --connection-id dxcon-fg9607vm
```

출력:

```
{
  "partnerName": "TIVIT",
  "vlan": 101,
  "ownerAccount": "123456789012",
  "connectionId": "dxcon-fg9607vm",
  "lagId": "dxlag-fhccu14t",
  "connectionState": "ordering",
  "bandwidth": "500Mbps",
  "location": "TIVIT",
  "connectionName": "mydcinterconnect",
  "region": "sa-east-1"
}
```

- 자세한 API 내용은 명령 참조AssociateHostedConnection의 섹션을 참조하세요. AWS CLI

### associate-virtual-interface

다음 코드 예시에서는 associate-virtual-interface을 사용하는 방법을 보여 줍니다.

AWS CLI

가상 인터페이스를 연결에 연결하려면

다음 예제에서는 지정된 가상 인터페이스를 지정된 와 연결합니다LAG. 또는 가상 인터페이스를 연결에 연결하려면 `--connection-id`와 같이 에 대한 AWS Direct Connect 연결의 ID를 지정합니다`dxcon-ffnikghc`.

명령:

```
aws directconnect associate-virtual-interface --connection-id dxlag-ffjhj9lx --
virtual-interface-id dxvif-fgputw0j
```

출력:

```
{
  "virtualInterfaceState": "pending",
  "asn": 65000,
  "vlan": 123,
  "customerAddress": "169.254.255.2/30",
  "ownerAccount": "123456789012",
  "connectionId": "dxlag-ffjhj9lx",
  "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"
}
```

- 자세한 API 내용은 명령 참조AssociateVirtualInterface의 섹션을 참조하세요. AWS CLI

### `confirm-connection`

다음 코드 예시에서는 `confirm-connection`을 사용하는 방법을 보여 줍니다.

AWS CLI

인터커넥트에서 호스팅 연결 생성을 확인하려면

다음 `confirm-connection` 명령은 상호 연결에서 호스팅된 연결이 생성되었음을 확인합니다.

```
aws directconnect confirm-connection --connection-id dxcon-fg2wi7hy
```

출력:

```
{
    "connectionState": "pending"
}
```

- 자세한 API 내용은 명령 참조ConfirmConnection의 섹션을 참조하세요. AWS CLI

### `confirm-private-virtual-interface`

다음 코드 예시에서는 `confirm-private-virtual-interface`을 사용하는 방법을 보여 줍니다.

AWS CLI

프라이빗 가상 인터페이스의 소유권을 수락하려면

다음 confirm-private-virtual-interface 명령은 다른 고객이 생성한 프라이빗 가상 인터페이스의 소유권을 허용합니다.

```
aws directconnect confirm-private-virtual-interface --virtual-interface-id dxvif-fgy8orxu --virtual-gateway-id vgw-e4a47df9
```

출력:

```
{
    "virtualInterfaceState": "pending"
}
```

- 자세한 API 내용은 명령 참조ConfirmPrivateVirtualInterface의 섹션을 참조하세요. AWS CLI

### confirm-public-virtual-interface

다음 코드 예시에서는 confirm-public-virtual-interface을 사용하는 방법을 보여 줍니다.

AWS CLI

퍼블릭 가상 인터페이스의 소유권을 수락하려면

다음 confirm-public-virtual-interface 명령은 다른 고객이 생성한 퍼블릭 가상 인터페이스의 소유권을 허용합니다.

```
aws directconnect confirm-public-virtual-interface --virtual-interface-id dxvif-fg9xo9vp
```

출력:

```
{
    "virtualInterfaceState": "verifying"
}
```

- 자세한 API 내용은 명령 참조ConfirmPublicVirtualInterface의 섹션을 참조하세요. AWS CLI

### confirm-transit-virtual-interface

다음 코드 예시에서는 confirm-transit-virtual-interface을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 가상 인터페이스의 소유권을 수락하려면

다음은 다른 고객이 생성한 전송 가상 인터페이스의 소유권을 `confirm-transit-virtual-interface` 허용합니다.

```
aws directconnect confirm-transit-virtual-interface \
    --virtual-interface-id dxvif-fEXAMPLE \
    --direct-connect-gateway-id 4112ccf9-25e9-4111-8237-b6c5dEXAMPLE
```

출력:

```
{
    "virtualInterfaceState": "pending"
}
```

자세한 내용은 AWS Direct Connect 사용 설명서의 호스팅 가상 인터페이스 수락을 참조하세요.

- 자세한 API 내용은 명령 참조ConfirmTransitVirtualInterface의 섹션을 참조하세요. AWS CLI

### `create-bgp-peer`

다음 코드 예시에서는 `create-bgp-peer`을 사용하는 방법을 보여 줍니다.

AWS CLI

IPv6 BGP 피어링 세션을 생성하려면

다음 예제에서는 프라이빗 가상 인터페이스 에서 IPv6 BGP 피어링 세션을 생성합니다`dxvif-fg1vuj3d`. 피어 IPv6 주소는 Amazon에서 자동으로 할당됩니다.

명령:

```
aws directconnect create-bgp-peer --virtual-interface-id dxvif-fg1vuj3d --new-bgp-peer asn=64600,addressFamily=ipv6
```

출력:

```
{
  "virtualInterface": {
      "virtualInterfaceState": "available",
```

```
      "asn": 65000,
      "vlan": 125,
      "customerAddress": "169.254.255.2/30",
      "ownerAccount": "123456789012",
      "connectionId": "dxcon-fguhmqlc",
      "addressFamily": "ipv4",
      "virtualGatewayId": "vgw-f9eb0c90",
      "virtualInterfaceId": "dxvif-fg1vuj3d",
      "authKey": "0xC_ukbCerl6EYA0example",
      "routeFilterPrefixes": [],
      "location": "EqDC2",
      "bgpPeers": [
          {
              "bgpStatus": "down",
              "customerAddress": "169.254.255.2/30",
              "addressFamily": "ipv4",
              "authKey": "0xC_ukbCerl6EYA0uexample",
              "bgpPeerState": "available",
              "amazonAddress": "169.254.255.1/30",
              "asn": 65000
          },
          {
              "bgpStatus": "down",
              "customerAddress": "2001:db8:1100:2f0:0:1:9cb4:4216/125",
              "addressFamily": "ipv6",
              "authKey": "0xS27kAIU_VHPjjAexample",
              "bgpPeerState": "pending",
              "amazonAddress": "2001:db8:1100:2f0:0:1:9cb4:4211/125",
              "asn": 64600
          }
      ],
      "customerRouterConfig": "<?xml version=\"1.0\" encoding=
\"UTF-8\"?>\n<logical_connection id=\"dxvif-fg1vuj3d\">\n  <vlan>125</
vlan>\n  <customer_address>169.254.255.2/30</customer_address>\n
 <amazon_address>169.254.255.1/30</amazon_address>\n  <bgp_asn>65000</
bgp_asn>\n  <bgp_auth_key>0xC_ukbCerl6EYA0uexample</bgp_auth_key>\n
 <ipv6_customer_address>2001:db8:1100:2f0:0:1:9cb4:4216/125</ipv6_customer_address>
\n  <ipv6_amazon_address>2001:db8:1100:2f0:0:1:9cb4:4211/125</ipv6_amazon_address>\n
  <ipv6_bgp_asn>64600</ipv6_bgp_asn>\n  <ipv6_bgp_auth_key>0xS27kAIU_VHPjjAexample</
ipv6_bgp_auth_key>\n  <amazon_bgp_asn>7224</amazon_bgp_asn>\n
 <connection_type>private</connection_type>\n</logical_connection>\n",
      "amazonAddress": "169.254.255.1/30",
      "virtualInterfaceType": "private",
      "virtualInterfaceName": "Test"
```

```
  }
}
```

- 자세한 API 내용은 명령 참조CreateBgpPeer의 섹션을 참조하세요. AWS CLI

### `create-connection`

다음 코드 예시에서는 `create-connection`을 사용하는 방법을 보여 줍니다.

AWS CLI

네트워크에서 AWS Direct Connect 위치로 연결을 생성하려면

다음 `create-connection` 명령은 네트워크에서 AWS Direct Connect 위치로 연결을 생성합니다.

```
aws directconnect create-connection --location TIVIT --bandwidth 1Gbps --connection-name "Connection to AWS"
```

출력:

```
{
    "ownerAccount": "123456789012",
    "connectionId": "dxcon-fg31dyv6",
    "connectionState": "requested",
    "bandwidth": "1Gbps",
    "location": "TIVIT",
    "connectionName": "Connection to AWS",
    "region": "sa-east-1"
}
```

- 자세한 API 내용은 명령 참조CreateConnection의 섹션을 참조하세요. AWS CLI

### `create-direct-connect-gateway-association-proposal`

다음 코드 예시에서는 `create-direct-connect-gateway-association-proposal`을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 전송 게이트웨이를 지정된 Direct Connect 게이트웨이와 연결하는 제안을 생성하려면

다음 create-direct-connect-gateway-association-proposal 예제에서는 지정된 전송 게이트웨이를 지정된 Direct Connect 게이트웨이와 연결하는 제안을 생성합니다.

```
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
```

출력:

```
{
    "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"
            }
        ]
    }
}
```

자세한 내용은 AWS Direct Connect 사용 설명서의 Transit Gateway 연결 제안 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateDirectConnectGatewayAssociationProposal의 섹션을 참조하세요. AWS CLI

### create-direct-connect-gateway-association

다음 코드 예시에서는 create-direct-connect-gateway-association을 사용하는 방법을 보여 줍니다.

AWS CLI

가상 프라이빗 게이트웨이를 Direct Connect 게이트웨이와 연결하려면

다음 예제에서는 가상 프라이빗 게이트웨이를 Direct Connect 게이트웨이 vgw-6efe725e와 연결합니다5f294f92-bafb-4011-916d-9b0bexample. 가상 프라이빗 게이트웨이가 위치한 리전에서 명령을 실행해야 합니다.

명령:

```
aws directconnect create-direct-connect-gateway-association --direct-connect-gateway-id 5f294f92-bafb-4011-916d-9b0bexample --virtual-gateway-id vgw-6efe725e
```

출력:

```
{
  "directConnectGatewayAssociation": {
      "associationState": "associating",
      "virtualGatewayOwnerAccount": "123456789012",
      "directConnectGatewayId": "5f294f92-bafb-4011-916d-9b0bexample",
      "virtualGatewayId": "vgw-6efe725e",
      "virtualGatewayRegion": "us-east-2"
  }
}
```

- 자세한 API 내용은 명령 참조CreateDirectConnectGatewayAssociation의 섹션을 참조하세요. AWS CLI

### create-direct-connect-gateway

다음 코드 예시에서는 create-direct-connect-gateway을 사용하는 방법을 보여 줍니다.

AWS CLI

Direct Connect 게이트웨이를 생성하려면

다음 예제에서는 이름이 인 Direct Connect 게이트웨이를 생성합니다DxGateway1.

명령:

```
aws directconnect create-direct-connect-gateway --direct-connect-gateway-name "DxGateway1"
```

출력:

```
{
  "directConnectGateway": {
      "amazonSideAsn": 64512,
      "directConnectGatewayId": "5f294f92-bafb-4011-916d-9b0bdexample",
      "ownerAccount": "123456789012",
      "directConnectGatewayName": "DxGateway1",
      "directConnectGatewayState": "available"
  }
}
```

- 자세한 API 내용은 명령 참조CreateDirectConnectGateway의 섹션을 참조하세요. AWS CLI

### `create-interconnect`

다음 코드 예시에서는 `create-interconnect`을 사용하는 방법을 보여 줍니다.

AWS CLI

파트너의 네트워크와 간의 상호 연결을 생성하려면 AWS

다음 `create-interconnect` 명령은 AWS Direct Connect 파트너의 네트워크와 특정 AWS Direct Connect 위치 간의 상호 연결을 생성합니다.

```
aws directconnect create-interconnect --interconnect-name "1G Interconnect to AWS"
 --bandwidth 1Gbps --location TIVIT
```

출력:

```
{
    "region": "sa-east-1",
    "bandwidth": "1Gbps",
    "location": "TIVIT",
    "interconnectName": "1G Interconnect to AWS",
    "interconnectId": "dxcon-fgktov66",
    "interconnectState": "requested"
}
```

- 자세한 API 내용은 명령 참조CreateInterconnect의 섹션을 참조하세요. AWS CLI

### create-lag

다음 코드 예시에서는 create-lag을 사용하는 방법을 보여 줍니다.

AWS CLI

새 연결LAG로 를 생성하려면

다음 예제에서는 를 생성하고 대역폭이 1GbpsLAG인 에 대한 두 개의 새 AWS Direct Connect 연결을 LAG 요청합니다.

명령:

```
aws directconnect create-lag --location CSVA1 --number-of-connections 2 --
connections-bandwidth 1Gbps --lag-name 1GBLag
```

출력:

```
{
  "awsDevice": "CSVA1-23u8tlpaz8iks",
  "numberOfConnections": 2,
  "lagState": "pending",
  "ownerAccount": "123456789012",
  "lagName": "1GBLag",
  "connections": [
      {
          "ownerAccount": "123456789012",
          "connectionId": "dxcon-ffqr6x5q",
          "lagId": "dxlag-ffjhj9lx",
          "connectionState": "requested",
          "bandwidth": "1Gbps",
          "location": "CSVA1",
          "connectionName": "Requested Connection 1 for Lag dxlag-ffjhj9lx",
          "region": "us-east-1"
      },
      {
          "ownerAccount": "123456789012",
          "connectionId": "dxcon-fflqyj95",
          "lagId": "dxlag-ffjhj9lx",
          "connectionState": "requested",
          "bandwidth": "1Gbps",
          "location": "CSVA1",
          "connectionName": "Requested Connection 2 for Lag dxlag-ffjhj9lx",
          "region": "us-east-1"
```

```
        }
    ],
    "lagId": "dxlag-ffjhj9lx",
    "minimumLinks": 0,
    "connectionsBandwidth": "1Gbps",
    "region": "us-east-1",
    "location": "CSVA1"
}
```

기존 연결을 LAG 사용하여 를 생성하려면

다음 예제에서는 계정의 기존 연결LAG에서 를 생성하고 기존 연결과 동일한 대역폭 및 위치를 LAG 가진 에 대한 두 번째 새 연결을 요청합니다.

명령:

```
aws directconnect create-lag --location EqDC2 --number-of-connections 2 --
connections-bandwidth 1Gbps --lag-name 2ConnLAG --connection-id dxcon-fgk145dr
```

출력:

```
{
    "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"
}
```

- 자세한 API 내용은 명령 참조CreateLag의 섹션을 참조하세요. AWS CLI

### `create-private-virtual-interface`

다음 코드 예시에서는 `create-private-virtual-interface`을 사용하는 방법을 보여 줍니다.

AWS CLI

프라이빗 가상 인터페이스를 생성하려면

다음 `create-private-virtual-interface` 명령은 프라이빗 가상 인터페이스를 생성합니다.

```
aws directconnect create-private-virtual-interface --connection-id dxcon-ffjrkx17 --
new-private-virtual-
interface virtualInterfaceName=PrivateVirtualInterface,vlan=101,asn=65000,authKey=asdf34exam
aba37db6
```

출력:

```
{
    "virtualInterfaceState": "pending",
    "asn": 65000,
    "vlan": 101,
    "customerAddress": "192.168.1.2/30",
    "ownerAccount": "123456789012",
    "connectionId": "dxcon-ffjrkx17",
    "virtualGatewayId": "vgw-aba37db6",
    "virtualInterfaceId": "dxvif-ffhhk74f",
    "authKey": "asdf34example",
    "routeFilterPrefixes": [],
```

```
    "location": "TIVIT",
    "customerRouterConfig": "<?xml version=\"1.0\" encoding=
\"UTF-8\"?>\n<logical_connection id=\"dxvif-ffhhk74f\">\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"
}
```

- 자세한 API 내용은 명령 참조CreatePrivateVirtualInterface의 섹션을 참조하세요. AWS CLI

### `create-public-virtual-interface`

다음 코드 예시에서는 `create-public-virtual-interface`을 사용하는 방법을 보여 줍니다.

AWS CLI

퍼블릭 가상 인터페이스를 생성하려면

다음 `create-public-virtual-interface` 명령은 퍼블릭 가상 인터페이스를 생성합니다.

```
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}]
```

출력:

```
{
    "virtualInterfaceState": "verifying",
    "asn": 65000,
    "vlan": 2000,
    "customerAddress": "203.0.113.2/30",
    "ownerAccount": "123456789012",
    "connectionId": "dxcon-ffjrkx17",
    "virtualInterfaceId": "dxvif-fgh0hcrk",
    "authKey": "asdf34example",
    "routeFilterPrefixes": [
        {
            "cidr": "203.0.113.0/30"
```

```
        },
        {
            "cidr": "203.0.113.4/30"
        }
    ],
    "location": "TIVIT",
    "customerRouterConfig": "<?xml version=\"1.0\" encoding=\"UTF-8\"?
>\n<logical_connection id=\"dxvif-fgh0hcrk\">\n  <vlan>2000</
vlan>\n  <customer_address>203.0.113.2/30</customer_address>\n
 <amazon_address>203.0.113.1/30</amazon_address>\n  <bgp_asn>65000</bgp_asn>\n
 <bgp_auth_key>asdf34example</bgp_auth_key>\n  <amazon_bgp_asn>7224</amazon_bgp_asn>
\n  <connection_type>public</connection_type>\n</logical_connection>\n",
    "amazonAddress": "203.0.113.1/30",
    "virtualInterfaceType": "public",
    "virtualInterfaceName": "PublicVirtualInterface"
}
```

- 자세한 API 내용은 명령 참조CreatePublicVirtualInterface의 섹션을 참조하세요. AWS CLI

### `create-transit-virtual-interface`

다음 코드 예시에서는 `create-transit-virtual-interface`을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 가상 인터페이스를 생성하려면

다음 `create-transit-virtual-interface` 예제에서는 지정된 연결을 위한 전송 가상 인터페이스를 생성합니다.

```
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}]"
```

출력:

```
{
    "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>0xzxgA9YoW9h58u8SvOmXRTw</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"
            }
```

```
        ]
    }
}
```

자세한 내용은 Direct Connect 사용 설명서의 Direct Connect Gateway에 대한 전송 가상 인터페이스 생성을 참조하세요. AWS

- 자세한 API 내용은 명령 참조CreateTransitVirtualInterface의 섹션을 참조하세요. AWS CLI

### `delete-bgp-peer`

다음 코드 예시에서는 `delete-bgp-peer`을 사용하는 방법을 보여 줍니다.

AWS CLI

가상 인터페이스에서 BGP 피어를 삭제하려면

다음 예제에서는 가상 인터페이스 에서 IPv6 BGP 피어를 삭제합니다`dxvif-fg1vuj3d`.

명령:

```
aws directconnect delete-bgp-peer --virtual-interface-id dxvif-fg1vuj3d --asn 64600
 --customer-address 2001:db8:1100:2f0:0:1:9cb4:4216/125
```

출력:

```
{
  "virtualInterface": {
      "virtualInterfaceState": "available",
      "asn": 65000,
      "vlan": 125,
      "customerAddress": "169.254.255.2/30",
      "ownerAccount": "123456789012",
      "connectionId": "dxcon-fguhmqlc",
      "addressFamily": "ipv4",
      "virtualGatewayId": "vgw-f9eb0c90",
      "virtualInterfaceId": "dxvif-fg1vuj3d",
      "authKey": "0xC_ukbCerl6EYA0example",
      "routeFilterPrefixes": [],
      "location": "EqDC2",
      "bgpPeers": [
          {
              "bgpStatus": "down",
```

```
                "customerAddress": "169.254.255.2/30",
                "addressFamily": "ipv4",
                "authKey": "0xC_ukbCerl6EYA0uexample",
                "bgpPeerState": "available",
                "amazonAddress": "169.254.255.1/30",
                "asn": 65000
            },
            {
                "bgpStatus": "down",
                "customerAddress": "2001:db8:1100:2f0:0:1:9cb4:4216/125",
                "addressFamily": "ipv6",
                "authKey": "0xS27kAIU_VHPjjAexample",
                "bgpPeerState": "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"
    }
}
```

- 자세한 API 내용은 명령 참조DeleteBgpPeer의 섹션을 참조하세요. AWS CLI

### `delete-connection`

다음 코드 예시에서는 `delete-connection`을 사용하는 방법을 보여 줍니다.

AWS CLI

연결을 삭제하려면

다음 `delete-connection` 명령은 지정된 연결을 삭제합니다.

```
aws directconnect delete-connection --connection-id dxcon-fg31dyv6
```

출력:

```
{
    "ownerAccount": "123456789012",
    "connectionId": "dxcon-fg31dyv6",
    "connectionState": "deleted",
    "bandwidth": "1Gbps",
    "location": "TIVIT",
    "connectionName": "Connection to AWS",
    "region": "sa-east-1"
}
```

- 자세한 API 내용은 명령 참조DeleteConnection의 섹션을 참조하세요. AWS CLI

### `delete-direct-connect-gateway-association`

다음 코드 예시에서는 `delete-direct-connect-gateway-association`을 사용하는 방법을 보여 줍니다.

AWS CLI

Direct Connect 게이트웨이 연결을 삭제하려면

다음 `delete-direct-connect-gateway-association` 예제에서는 지정된 연결 ID가 있는 전송 게이트웨이와의 Direct Connect 게이트웨이 연결을 삭제합니다.

```
aws directconnect delete-direct-connect-gateway-association --association-id
  be85116d-46eb-4b43-a27a-da0c2ad648de
```

출력:

```
{
    "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"
            }
        ]
    }
}
```

자세한 내용은 AWS Direct Connect 사용 설명서의 Transit Gateway 연결 및 연결 해제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteDirectConnectGatewayAssociation의 섹션을 참조하세요. AWS CLI

### `delete-direct-connect-gateway`

다음 코드 예시에서는 `delete-direct-connect-gateway`을 사용하는 방법을 보여 줍니다.

AWS CLI

Direct Connect 게이트웨이를 삭제하려면

다음 예제에서는 Direct Connect 게이트웨이 를 삭제합니다`5f294f92-bafb-4011-916d-9b0bexample`.

명령:

```
aws directconnect delete-direct-connect-gateway --direct-connect-gateway-
id 5f294f92-bafb-4011-916d-9b0bexample
```

출력:

```
{
  "directConnectGateway": {
      "amazonSideAsn": 64512,
      "directConnectGatewayId": "5f294f92-bafb-4011-916d-9b0bexample",
      "ownerAccount": "123456789012",
      "directConnectGatewayName": "DxGateway1",
      "directConnectGatewayState": "deleting"
  }
}
```

- 자세한 API 내용은 명령 참조DeleteDirectConnectGateway의 섹션을 참조하세요. AWS CLI

### delete-interconnect

다음 코드 예시에서는 delete-interconnect을 사용하는 방법을 보여 줍니다.

AWS CLI

상호 연결을 삭제하려면

다음 delete-interconnect 명령은 지정된 상호 연결을 삭제합니다.

```
aws directconnect delete-interconnect --interconnect-id dxcon-fgktov66
```

출력:

```
{
    "interconnectState": "deleted"
}
```

- 자세한 API 내용은 명령 참조DeleteInterconnect의 섹션을 참조하세요. AWS CLI

### delete-lag

다음 코드 예시에서는 delete-lag을 사용하는 방법을 보여 줍니다.

AWS CLI

를 삭제하려면 LAG

다음 예제에서는 지정된 를 삭제합니다LAG.

명령:

```
aws directconnect delete-lag --lag-id dxlag-ffrhowd9
```

출력:

```
{
  "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"
}
```

- 자세한 API 내용은 명령 참조DeleteLag의 섹션을 참조하세요. AWS CLI

### `delete-virtual-interface`

다음 코드 예시에서는 `delete-virtual-interface`을 사용하는 방법을 보여 줍니다.

AWS CLI

가상 인터페이스를 삭제하려면

다음 `delete-virtual-interface` 명령은 지정된 가상 인터페이스를 삭제합니다.

```
aws directconnect delete-virtual-interface --virtual-interface-id dxvif-ffhhk74f
```

출력:

```
{
    "virtualInterfaceState": "deleting"
}
```

- 자세한 API 내용은 명령 참조DeleteVirtualInterface의 섹션을 참조하세요. AWS CLI

### `describe-connection-loa`

다음 코드 예시에서는 `describe-connection-loa`을 사용하는 방법을 보여 줍니다.

AWS CLI

Linux 또는 Mac OS X를 사용한 연결에 대한 LOA를CFA 설명하려면

다음 예제에서는 연결용 LOA-CFA에 대해 설명합니다`dxcon-fh6ayh1d`. LOA-CFA의 내용은 base64로 인코딩됩니다. 이 명령은 `--output` 및 `--query` 파라미터를 사용하여 출력을 제어하고

loaContent 구조의 내용을 추출합니다. 명령의 마지막 부분은 base64 유틸리티를 사용하여 콘텐츠를 디코딩하고 출력을 PDF 파일로 전송합니다.

```
aws directconnect describe-connection-loa --connection-id dxcon-fh6ayh1d --
output text --query loa.loaContent|base64 --decode > myLoaCfa.pdf
```

Windows를 사용한 연결에 대한 LOA-CFA 설명

이전 예제에서는 base64 유틸리티를 사용하여 출력을 디코딩해야 합니다. Windows 컴퓨터에서certutil는 대신 를 사용할 수 있습니다. 다음 예제에서 첫 번째 명령은 LOA-CFA for connection을 설명하고 dxcon-fh6ayh1d 및 --output --query 파라미터를 사용하여 출력을 제어하고 loaContent 구조의 내용을 라는 파일에 추출합니다myLoaCfa.base64. 두 번째 명령은 certutil 유틸리티를 사용하여 파일을 디코딩하고 출력을 PDF 파일로 전송합니다.

```
aws directconnect describe-connection-loa --connection-id dxcon-fh6ayh1d --
output text --query loa.loaContent > myLoaCfa.base64
```

```
certutil -decode myLoaCfa.base64 myLoaCfa.pdf
```

출력 제어 AWS CLI에 대한 자세한 내용은 명령줄 인터페이스 사용 설명서의 AWS 명령줄 인터페이스에서 명령 출력 제어를 참조하세요. AWS

- 자세한 API 내용은 명령 참조DescribeConnectionLoa의 섹션을 참조하세요. AWS CLI

### describe-connections-on-interconnect

다음 코드 예시에서는 describe-connections-on-interconnect을 사용하는 방법을 보여 줍니다.

AWS CLI

상호 연결의 연결을 나열하려면

다음 describe-connections-on-interconnect 명령은 지정된 상호 연결에서 프로비저닝된 연결을 나열합니다.

```
aws directconnect describe-connections-on-interconnect --interconnect-id dxcon-
fgktov66
```

출력:

```
{
    "connections": [
        {
            "partnerName": "TIVIT",
            "vlan": 101,
            "ownerAccount": "123456789012",
            "connectionId": "dxcon-ffzc51m1",
            "connectionState": "ordering",
            "bandwidth": "500Mbps",
            "location": "TIVIT",
            "connectionName": "mydcinterconnect",
            "region": "sa-east-1"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeConnectionsOnInterconnect의 섹션을 참조하세요. AWS CLI

### `describe-connections`

다음 코드 예시에서는 `describe-connections`을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 리전의 모든 연결을 나열하려면

다음 `describe-connections` 명령은 현재 리전의 모든 연결을 나열합니다.

```
aws directconnect describe-connections
```

출력:

```
{
  "connections": [
      {
          "awsDevice": "EqDC2-123h49s71dabc",
          "ownerAccount": "123456789012",
          "connectionId": "dxcon-fguhmqlc",
          "lagId": "dxlag-ffrz71kw",
          "connectionState": "down",
          "bandwidth": "1Gbps",
```

```
            "location": "EqDC2",
            "connectionName": "My_Connection",
            "loaIssueTime": 1491568964.0,
            "region": "us-east-1"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeConnections의 섹션을 참조하세요. AWS CLI

### `describe-direct-connect-gateway-association-proposals`

다음 코드 예시에서는 `describe-direct-connect-gateway-association-proposals`을 사용하는 방법을 보여 줍니다.

AWS CLI

Direct Connect 게이트웨이 연결 제안을 설명하려면

다음 `describe-direct-connect-gateway-association-proposals` 예제에서는 Direct Connect 게이트웨이 연결 제안에 대한 세부 정보를 보여줍니다.

```
aws directconnect describe-direct-connect-gateway-association-proposals
```

출력:

```
{
    "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"
                }
            ]
        }
    ]
}
```

자세한 내용은 AWS Direct Connect 사용 설명서의 전송 게이트웨이 연결 및 연결 해제를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeDirectConnectGatewayAssociationProposals의 섹션을 참조하세요. AWS CLI

### describe-direct-connect-gateway-associations

다음 코드 예시에서는 describe-direct-connect-gateway-associations을 사용하는 방법을 보여 줍니다.

AWS CLI

Direct Connect 게이트웨이 연결을 설명하려면

다음 예제에서는 Direct Connect 게이트웨이와의 모든 연결을 설명합니다5f294f92-bafb-4011-916d-9b0bexample.

명령:

```
aws directconnect describe-direct-connect-gateway-associations --direct-connect-gateway-id 5f294f92-bafb-4011-916d-9b0bexample
```

출력:

```
{
  "nextToken":
 "eyJ2IjoxLCJzIjoxLCJpIjoiOU83OTFodzdycnZCbkN4MExHeHVwQT09IiwiYyI6InIxTEN0UEVHV0I1UFlkaWFnNl
  "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"
      }
  ]
}
```

- 자세한 API 내용은 명령 참조DescribeDirectConnectGatewayAssociations의 섹션을 참조하세요. AWS CLI

### describe-direct-connect-gateway-attachments

다음 코드 예시에서는 describe-direct-connect-gateway-attachments을 사용하는 방법을 보여 줍니다.

AWS CLI

Direct Connect 게이트웨이 연결 설명

다음 예제에서는 Direct Connect 게이트웨이 에 연결된 가상 인터페이스를 설명합니다5f294f92-bafb-4011-916d-9b0bexample.

명령:

```
aws directconnect describe-direct-connect-gateway-attachments --direct-connect-gateway-id 5f294f92-bafb-4011-916d-9b0bexample
```

출력:

```
{
  "directConnectGatewayAttachments": [
      {
          "virtualInterfaceOwnerAccount": "123456789012",
          "directConnectGatewayId": "5f294f92-bafb-4011-916d-9b0bexample",
          "virtualInterfaceRegion": "us-east-2",
          "attachmentState": "attaching",
          "virtualInterfaceId": "dxvif-fg9zyabc"
      }
  ],
  "nextToken":
 "eyJ2IjoxLCJzIjoxLCJpIjoibEhXdlNpUXF5RzhoL1JyUW52SlV2QT09IiwiYyI6Im5wQjFHQ0RyQUdRS3puNnNXcU
}
```

- 자세한 API 내용은 명령 참조DescribeDirectConnectGatewayAttachments의 섹션을 참조하세요. AWS CLI

### describe-direct-connect-gateways

다음 코드 예시에서는 describe-direct-connect-gateways을 사용하는 방법을 보여 줍니다.

AWS CLI

Direct Connect 게이트웨이를 설명하려면

다음 예제에서는 모든 Direct Connect 게이트웨이를 설명합니다.

명령:

```
aws directconnect describe-direct-connect-gateways
```

출력:

```
{
  "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"
      }
  ]
}
```

- 자세한 API 내용은 명령 참조DescribeDirectConnectGateways의 섹션을 참조하세요. AWS CLI

### describe-hosted-connections

다음 코드 예시에서는 describe-hosted-connections을 사용하는 방법을 보여 줍니다.

AWS CLI

상호 연결의 연결을 나열하려면

다음 예제에서는 지정된 인터커넥트에서 프로비저닝된 연결을 나열합니다.

명령:

```
aws directconnect describe-hosted-connections --connection-id dxcon-fgktov66
```

출력:

```
{
    "connections": [
        {
            "partnerName": "TIVIT",
            "vlan": 101,
            "ownerAccount": "123456789012",
            "connectionId": "dxcon-ffzc51m1",
            "connectionState": "ordering",
            "bandwidth": "500Mbps",
            "location": "TIVIT",
            "connectionName": "mydcinterconnect",
            "region": "sa-east-1"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeHostedConnections의 섹션을 참조하세요. AWS CLI

### `describe-interconnect-loa`

다음 코드 예시에서는 `describe-interconnect-loa`을 사용하는 방법을 보여 줍니다.

AWS CLI

Linux 또는 Mac OS X를 사용한 상호 연결에 대한 LOA를CFA 설명하려면

다음 예제에서는 상호 연결용 LOA-CFA 에 대해 설명합니다`dxcon-fh6ayh1d`. LOA-CFA의 내용은 base64로 인코딩됩니다. 이 명령은 `--output` 및 `--query` 파라미터를 사용하여 출력을 제어하고 `loaContent` 구조의 내용을 추출합니다. 명령의 마지막 부분은 `base64` 유틸리티를 사용하여 콘텐츠를 디코딩하고 출력을 PDF 파일로 전송합니다.

```
aws directconnect describe-interconnect-loa --interconnect-id dxcon-fh6ayh1d --
output text --query loa.loaContent|base64 --decode > myLoaCfa.pdf
```

Windows를 사용한 상호 연결에 대한 LOA-CFA 설명

이전 예제에서는 base64 유틸리티를 사용하여 출력을 디코딩해야 합니다. Windows 컴퓨터에서certutil는 대신 를 사용할 수 있습니다. 다음 예제에서 첫 번째 명령은 LOA-CFA for interconnect를 설명하고 dxcon-fh6ayh1d 및 --output --query 파라미터를 사용하여 출력을 제어하고 loaContent 구조의 내용을 라는 파일에 추출합니다myLoaCfa.base64. 두 번째 명령은 certutil 유틸리티를 사용하여 파일을 디코딩하고 출력을 PDF 파일로 전송합니다.

```
aws directconnect describe-interconnect-loa --interconnect-id dxcon-fh6ayh1d --
output text --query loa.loaContent > myLoaCfa.base64
```

```
certutil -decode myLoaCfa.base64 myLoaCfa.pdf
```

출력 제어 AWS CLI에 대한 자세한 내용은 명령줄 인터페이스 사용 설명서의 AWS 명령줄 인터페이스에서 명령 출력 제어를 참조하세요. AWS

- 자세한 API 내용은 명령 참조DescribeInterconnectLoa의 섹션을 참조하세요. AWS CLI

### describe-interconnects

다음 코드 예시에서는 describe-interconnects을 사용하는 방법을 보여 줍니다.

AWS CLI

상호 연결을 나열하려면

다음 describe-interconnects 명령은 AWS 계정에서 소유한 인터커넥트를 나열합니다.

```
aws directconnect describe-interconnects
```

출력:

```
{
    "interconnects": [
        {
            "region": "sa-east-1",
            "bandwidth": "1Gbps",
            "location": "TIVIT",
            "interconnectName": "1G Interconnect to AWS",
            "interconnectId": "dxcon-fgktov66",
            "interconnectState": "down"
        }
    ]
```

```
}
```

- 자세한 API 내용은 명령 참조DescribeInterconnects의 섹션을 참조하세요. AWS CLI

### `describe-lags`

다음 코드 예시에서는 `describe-lags`을 사용하는 방법을 보여 줍니다.

AWS CLI

를 설명하려면 LAGs

다음 명령은 현재 리전의 모든 LAGs 를 설명합니다.

명령:

```
aws directconnect describe-lags
```

출력:

```
{
  "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"
    }
  ]
}
```

- 자세한 API 내용은 명령 참조DescribeLags의 섹션을 참조하세요. AWS CLI

### `describe-loa`

다음 코드 예시에서는 `describe-loa`을 사용하는 방법을 보여 줍니다.

AWS CLI

Linux 또는 Mac OS X를 사용한 연결에 대한 LOA-CFA 설명

다음 예제에서는 연결용 LOA-CFA에 대해 설명합니다`dxcon-fh6ayh1d`. LOA-CFA의 내용은 base64로 인코딩됩니다. 이 명령은 `--output` 및 `--query` 파라미터를 사용하여 출력을 제어하고 `loaContent` 구조의 내용을 추출합니다. 명령의 마지막 부분은 `base64` 유틸리티를 사용하여 콘텐츠를 디코딩하고 출력을 PDF 파일로 전송합니다.

```
aws directconnect describe-loa --connection-id dxcon-fh6ayh1d --output text --
query loa.loaContent|base64 --decode > myLoaCfa.pdf
```

Windows를 사용한 연결에 대한 LOA-CFA 설명

이전 예제에서는 `base64` 유틸리티를 사용하여 출력을 디코딩해야 합니다. Windows 컴퓨터에서`certutil`는 대신 를 사용할 수 있습니다. 다음 예제에서 첫 번째 명령은 LOA-CFA for connection을 설명하고 `dxcon-fh6ayh1d` 및 `--output` `--query` 파라미터를 사용하여 출력을 제어하고 `loaContent` 구조의 내용을 라는 파일에 추출합니다`myLoaCfa.base64`. 두 번째 명령은 `certutil` 유틸리티를 사용하여 파일을 디코딩하고 출력을 PDF 파일로 전송합니다.

```
aws directconnect describe-loa --connection-id dxcon-fh6ayh1d --output text --
query loa.loaContent > myLoaCfa.base64
```

```
certutil -decode myLoaCfa.base64 myLoaCfa.pdf
```

출력 제어 AWS CLI에 대한 자세한 내용은 명령줄 인터페이스 사용 설명서의 AWS 명령줄 인터페이스에서 명령 출력 제어를 참조하세요. AWS

- 자세한 API 내용은 명령 참조DescribeLoa의 섹션을 참조하세요. AWS CLI

### describe-locations

다음 코드 예시에서는 describe-locations을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS Direct Connect 파트너 및 위치를 나열하려면

다음 describe-locations 명령은 현재 리전의 AWS Direct Connect 파트너 및 위치를 나열합니다.

```
aws directconnect describe-locations
```

출력:

```
{
    "locations": [
        {
            "locationName": "NAP do Brasil, Barueri, Sao Paulo",
            "locationCode": "TNDB"
        },
        {
            "locationName": "Tivit - Site Transamerica (Sao Paulo)",
            "locationCode": "TIVIT"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeLocations의 섹션을 참조하세요. AWS CLI

### describe-tags

다음 코드 예시에서는 describe-tags을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS Direct Connect 리소스에 대한 태그를 설명하려면

다음 명령은 연결 에 대한 태그를 설명합니다dxcon-abcabc12.

명령:

```
aws directconnect describe-tags --resource-arns arn:aws:directconnect:us-
east-1:123456789012:dxcon/dxcon-abcabc12
```

출력:

```
{
  "resourceTags": [
      {
          "resourceArn": "arn:aws:directconnect:us-east-1:123456789012:dxcon/dxcon-
abcabc12",
          "tags": [
              {
                  "value": "VAConnection",
                  "key": "Name"
              }
          ]
      }
  ]
}
```

- 자세한 API 내용은 명령 참조DescribeTags의 섹션을 참조하세요. AWS CLI

### describe-virtual-gateways

다음 코드 예시에서는 describe-virtual-gateways을 사용하는 방법을 보여 줍니다.

AWS CLI

가상 프라이빗 게이트웨이를 나열하려면

다음 describe-virtual-gateways 명령은 AWS 계정에서 소유한 가상 프라이빗 게이트웨이를 나열합니다.

```
aws directconnect describe-virtual-gateways
```

출력:

```
{
    "virtualGateways": [
        {
            "virtualGatewayId": "vgw-aba37db6",
            "virtualGatewayState": "available"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeVirtualGateways의 섹션을 참조하세요. AWS CLI

### describe-virtual-interfaces

다음 코드 예시에서는 describe-virtual-interfaces을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 가상 인터페이스를 나열하려면

다음 describe-virtual-interfaces 명령은 AWS 계정과 연결된 모든 가상 인터페이스에 대한 정보를 나열합니다.

```
aws directconnect describe-virtual-interfaces --connection-id dxcon-ffjrkx17
```

출력:

```
{
    "virtualInterfaces": [
        {
            "virtualInterfaceState": "down",
            "asn": 65000,
            "vlan": 101,
            "customerAddress": "192.168.1.2/30",
            "ownerAccount": "123456789012",
            "connectionId": "dxcon-ffjrkx17",
```

```
            "virtualGatewayId": "vgw-aba37db6",
            "virtualInterfaceId": "dxvif-ffhhk74f",
            "authKey": "asdf34example",
            "routeFilterPrefixes": [],
            "location": "TIVIT",
            "customerRouterConfig": "<?xml version=\"1.0\" encoding=
\"UTF-8\"?>\n<logical_connection id=\"dxvif-ffhhk74f\">\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"
        }
```

```
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeVirtualInterfaces의 섹션을 참조하세요. AWS CLI

### `disassociate-connection-from-lag`

다음 코드 예시에서는 `disassociate-connection-from-lag`을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 연결을 해제하려면 LAG

다음 예제에서는 지정된 에서 지정된 연결을 연결 해제합니다LAG.

명령:

```
aws directconnect disassociate-connection-from-lag --lag-id dxlag-fhccu14t --
connection-id  dxcon-fg9607vm
```

출력:

```
{
  "ownerAccount": "123456789012",
  "connectionId": "dxcon-fg9607vm",
  "connectionState": "requested",
  "bandwidth": "1Gbps",
  "location": "EqDC2",
  "connectionName": "Con2ForLag",
  "region": "us-east-1"
}
```

- 자세한 API 내용은 명령 참조DisassociateConnectionFromLag의 섹션을 참조하세요. AWS CLI

### `tag-resource`

다음 코드 예시에서는 `tag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS Direct Connect 리소스에 태그를 추가하려면

다음 명령은 키 Name 와 값이 인 태그를 연결 VAConnection에 추가합니다dxcon-abcabc12. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws directconnect tag-resource --resource-arn arn:aws:directconnect:us-
east-1:123456789012:dxcon/dxcon-abcabc12 --tags "key=Name,value=VAConnection"
```

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### untag-resource

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS Direct Connect 리소스에서 태그를 제거하려면

다음 명령은 연결 Name에서 키가 있는 태그를 제거합니다dxcon-abcabc12. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws directconnect untag-resource --resource-arn arn:aws:directconnect:us-
east-1:123456789012:dxcon/dxcon-abcabc12 --tag-keys Name
```

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-direct-connect-gateway-association

다음 코드 예시에서는 update-direct-connect-gateway-association을 사용하는 방법을 보여 줍니다.

AWS CLI

Direct Connect 게이트웨이 연결의 지정된 속성을 업데이트하려면

다음 update-direct-connect-gateway-association 예제에서는 지정된 CIDR 블록을 Direct Connect 게이트웨이 연결에 추가합니다.

```
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
```

출력:

```
{
    "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"
            }
        ]
    }
}
```

자세한 내용은 Direct Connect 사용 설명서의 Direct Connect Gateway 작업을 참조하세요. AWS

- 자세한 API 내용은 명령 참조UpdateDirectConnectGatewayAssociation의 섹션을 참조하세요. AWS CLI

### update-lag

다음 코드 예시에서는 update-lag을 사용하는 방법을 보여 줍니다.

AWS CLI

를 업데이트하려면 LAG

다음 예제에서는 지정된 의 이름을 변경합니다LAG.

명령:

```
aws directconnect update-lag --lag-id dxlag-ffjhj9lx --lag-name 2ConnLag
```

출력:

```
{
  "awsDevice": "CSVA1-23u8tlpaz8iks",
  "numberOfConnections": 2,
  "lagState": "down",
  "ownerAccount": "123456789012",
  "lagName": "2ConnLag",
  "connections": [
      {
          "ownerAccount": "123456789012",
          "connectionId": "dxcon-fflqyj95",
          "lagId": "dxlag-ffjhj9lx",
          "connectionState": "requested",
          "bandwidth": "1Gbps",
          "location": "CSVA1",
          "connectionName": "Requested Connection 2 for Lag dxlag-ffjhj9lx",
          "region": "us-east-1"
      },
      {
          "ownerAccount": "123456789012",
          "connectionId": "dxcon-ffqr6x5q",
          "lagId": "dxlag-ffjhj9lx",
          "connectionState": "requested",
          "bandwidth": "1Gbps",
          "location": "CSVA1",
          "connectionName": "Requested Connection 1 for Lag dxlag-ffjhj9lx",
          "region": "us-east-1"
      }
  ],
  "lagId": "dxlag-ffjhj9lx",
  "minimumLinks": 0,
  "connectionsBandwidth": "1Gbps",
  "region": "us-east-1",
  "location": "CSVA1"
}
```

- 자세한 API 내용은 명령 참조UpdateLag의 섹션을 참조하세요. AWS CLI

### update-virtual-interface-attributes

다음 코드 예시에서는 update-virtual-interface-attributes을 사용하는 방법을 보여 줍니다.

AWS CLI

가상 인터페이스MTU의 를 업데이트하려면

다음 update-virtual-interface-attributes 예제에서는 지정된 가상 인터페이스MTU의 를 업데이트합니다.

```
aws directconnect update-virtual-interface-attributes \
    --virtual-interface-id dxvif-fEXAMPLE \
    --mtu 1500
```

출력:

```
{
    "ownerAccount": "1111222233333",
    "virtualInterfaceId": "dxvif-fEXAMPLE",
    "location": "loc1",
    "connectionId": "dxlag-fEXAMPLE",
    "virtualInterfaceType": "transit",
    "virtualInterfaceName": "example transit virtual interface",
    "vlan": 125,
    "asn": 650001,
    "amazonSideAsn": 64512,
    "authKey": "0xzxgA9YoW9h58u8SEXAMPLE",
    "amazonAddress": "169.254.248.1/30",
    "customerAddress": "169.254.248.2/30",
    "addressFamily": "ipv4",
    "virtualInterfaceState": "down",
    "customerRouterConfig": "<?xml version=\"1.0\" encoding=\"UTF-8\"?
>\n<logical_connection id=\"dxvif-fEXAMPLE\">\n  <vlan>125</vlan>
\n  <customer_address>169.254.248.2/30</customer_address>\n
 <amazon_address>169.254.248.1/30</amazon_address>\n  <bgp_asn>650001</bgp_asn>\n
  <bgp_auth_key>0xzxgA9YoW9h58u8SEXAMPLE</bgp_auth_key>\n  <amazon_bgp_asn>64512</
amazon_bgp_asn>\n  <connection_type>transit</connection_type>\n</logical_connection>
\n",
    "mtu": 1500,
    "jumboFrameCapable": true,
```

```
    "virtualGatewayId": "",
    "directConnectGatewayId": "879b76a1-403d-4700-8b53-4a56ed85436e",
    "routeFilterPrefixes": [],
    "bgpPeers": [
        {
            "bgpPeerId": "dxpeer-fEXAMPLE",
            "asn": 650001,
            "authKey": "0xzxgA9YoW9h58u8SEXAMPLE",
            "addressFamily": "ipv4",
            "amazonAddress": "169.254.248.1/30",
            "customerAddress": "169.254.248.2/30",
            "bgpPeerState": "available",
            "bgpStatus": "down",
            "awsDeviceV2": "loc1-26wz6vEXAMPLE"
        }
    ],
    "region": "sa-east-1",
    "awsDeviceV2": "loc1-26wz6vEXAMPLE",
    "tags": []
}
```

자세한 내용은 AWS Direct Connect 사용 설명서의 프라이빗 가상 인터페이스MTU용 네트워크 설정 또는 가상 인터페이스 전송을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateVirtualInterfaceAttributes의 섹션을 참조하세요. AWS CLI

## AWS Directory Service 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS Directory Service.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### describe-directories

다음 코드 예시에서는 describe-directories을 사용하는 방법을 보여 줍니다.

AWS CLI

디렉터리에 대한 세부 정보를 가져오려면

다음 describe-directories 예제에서는 지정된 디렉터리에 대한 세부 정보를 표시합니다.

```
aws ds describe-directories \
   --directory-id d-a1b2c3d4e5
```

출력:

```
{
    "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"
                    ]
                }
            }
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeDirectories의 섹션을 참조하세요. AWS CLI

### describe-trusts

다음 코드 예시에서는 describe-trusts을 사용하는 방법을 보여 줍니다.

AWS CLI

신뢰 관계에 대한 세부 정보를 가져오려면

다음 describe-trusts 예제에서는 지정된 디렉터리의 신뢰 관계에 대한 세부 정보를 표시합니다.

```
aws ds describe-trusts \
   --directory-id d-a1b2c3d4e5
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeTrusts의 섹션을 참조하세요. AWS CLI

## AWS DMS 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS DMS.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `add-tags-to-resource`

다음 코드 예시에서는 `add-tags-to-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 태그를 추가하려면

다음 `add-tags-to-resource` 예제에서는 복제 인스턴스에 태그를 추가합니다.

```
aws dms add-tags-to-resource \
    --resource-arn arn:aws:dms:us-east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE
 \
    --tags Key=Environment,Value=PROD Key=Project,Value=dbMigration
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS 데이터베이스 마이그레이션 서비스 사용 설명서의 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조AddTagsToResource의 섹션을 참조하세요. AWS CLI

### `create-endpoint`

다음 코드 예시에서는 `create-endpoint`을 사용하는 방법을 보여 줍니다.

AWS CLI

엔드포인트를 생성하려면

다음 `create-endpoint` 예제에서는 Amazon S3 소스에 대한 엔드포인트를 생성합니다.

```
aws dms create-endpoint \
    --endpoint-type source \
    --engine-name s3 \
    --endpoint-identifier src-endpoint \
    --s3-settings file://s3-settings.json
```

`s3-settings.json`의 콘텐츠:

```
{
    "BucketName":"my-corp-data",
    "BucketFolder":"sourcedata",
    "ServiceAccessRoleArn":"arn:aws:iam::123456789012:role/my-s3-access-role"
}
```

출력:

```
{
    "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:GUVAFG34EECUOJ6QVZ56DAHT3U",
        "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
        }
    }
}
```

자세한 내용은 AWS 데이터베이스 마이그레이션 서비스 사용 설명서의 엔드포인트 작업을 AWS DMS 참조하세요.

- 자세한 API 내용은 명령 참조CreateEndpoint의 섹션을 참조하세요. AWS CLI

### `create-event-subscription`

다음 코드 예시에서는 `create-event-subscription`을 사용하는 방법을 보여 줍니다.

AWS CLI

이벤트 구독을 나열하려면

다음 `create-event-subscription` 예제에서는 Amazon SNS 주제()에 대한 이벤트 구독을 생성합니다my-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
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 AWS 데이터베이스 마이그레이션 서비스 사용 설명서의 이벤트 및 알림 작업을 참조하세요.

- 자세한 API 내용은 명령 참조CreateEventSubscription의 섹션을 참조하세요. AWS CLI

### `create-replication-instance`

다음 코드 예시에서는 `create-replication-instance`을 사용하는 방법을 보여 줍니다.

AWS CLI

복제 인스턴스를 생성하려면

다음 `create-replication-instance` 예제에서는 복제 인스턴스를 생성합니다.

```
aws dms create-replication-instance \
    --replication-instance-identifier my-repl-instance \
    --replication-instance-class dms.t2.micro \
    --allocated-storage 5
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 AWS 데이터베이스 마이그레이션 서비스 사용 설명서의 복제 인스턴스 작업을 AWS DMS 참조하세요.

- 자세한 API 내용은 명령 참조CreateReplicationInstance의 섹션을 참조하세요. AWS CLI

### create-replication-subnet-group

다음 코드 예시에서는 create-replication-subnet-group을 사용하는 방법을 보여 줍니다.

AWS CLI

서브넷 그룹을 생성하려면

다음 create-replication-subnet-group 예제에서는 3개의 서브넷으로 구성된 그룹을 생성합니다.

```
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
```

출력:

```
{
```

```
    "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"
            }
        ]
    }
}
```

자세한 내용은 AWS 데이터베이스 마이그레이션 서비스 사용 설명서의 복제 인스턴스에 대한 네트워크 설정을 참조하세요.

- 자세한 API 내용은 명령 참조CreateReplicationSubnetGroup의 섹션을 참조하세요. AWS CLI

### `create-replication-task`

다음 코드 예시에서는 `create-replication-task`을 사용하는 방법을 보여 줍니다.

AWS CLI

복제 작업을 생성하려면

다음 create-replication-task 예제에서는 복제 작업을 생성합니다.

```
aws dms create-replication-task \
    --replication-task-identifier movedata \
    --source-endpoint-arn arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA \
    --target-endpoint-arn arn:aws:dms:us-
east-1:123456789012:endpoint:EOM4SFKCZEYHZBFGAGZT3QEC5U \
    --replication-instance-arn $RI_ARN \
    --migration-type full-load \
    --table-mappings file://table-mappings.json
```

table-mappings.json의 콘텐츠:

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "prodrep",
                "table-name": "%"
            },
            "rule-action": "include",
            "filters": []
        }
    ]
}
```

출력:

```
{
    "ReplicationTask": {
        "ReplicationTaskIdentifier": "moveit2",
        "SourceEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
        "TargetEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:EOM4SFKCZEYHZBFGAGZT3QEC5U",
        "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE",
        "MigrationType": "full-load",
```

```
        "TableMappings": ...output omitted... ,
        "ReplicationTaskSettings": ...output omitted... ,
        "Status": "creating",
        "ReplicationTaskCreationDate": 1590524772.505,
        "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"
    }
}
```

자세한 내용은 AWS 데이터베이스 마이그레이션 서비스 사용 설명서의 작업 작업을 AWS DMS 참조하세요.

- 자세한 API 내용은 명령 참조CreateReplicationTask의 섹션을 참조하세요. AWS CLI

### `delete-connection`

다음 코드 예시에서는 `delete-connection`을 사용하는 방법을 보여 줍니다.

AWS CLI

연결을 삭제하려면

다음 `delete-connection` 예제에서는 복제 인스턴스에서 엔드포인트의 연결을 해제합니다.

```
aws dms delete-connection \
    --endpoint-arn arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA \
    --replication-instance-arn arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE
```

출력:

```
{
    "Connection": {
        "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE",
        "EndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
        "Status": "deleting",
        "EndpointIdentifier": "src-database-1",
        "ReplicationInstanceIdentifier": "my-repl-instance"
    }
```

```
}
```

자세한 내용은 AWS 데이터베이스 마이그레이션 서비스 사용 설명서의 https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Endpoints.Creating.html을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteConnection의 섹션을 참조하세요. AWS CLI

### delete-endpoint

다음 코드 예시에서는 delete-endpoint을 사용하는 방법을 보여 줍니다.

AWS CLI

엔드포인트를 삭제하려면

다음 delete-endpoint 예제에서는 엔드포인트를 삭제합니다.

```
aws dms delete-endpoint \
    --endpoint-arn arn:aws:dms:us-
east-1:123456789012:endpoint:OUJJVXO4XZ4CYTSEG5XGMN2R3Y
```

출력:

```
{
    "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:OUJJVXO4XZ4CYTSEG5XGMN2R3Y",
        "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
        }
    }
}
```

자세한 내용은 AWS 데이터베이스 마이그레이션 서비스 사용 설명서의 엔드포인트 작업을 AWS DMS 참조하세요.

- 자세한 API 내용은 명령 참조DeleteEndpoint의 섹션을 참조하세요. AWS CLI

### `delete-event-subscription`

다음 코드 예시에서는 `delete-event-subscription`을 사용하는 방법을 보여 줍니다.

AWS CLI

이벤트 구독을 삭제하려면

다음 `delete-event-subscription` 예제에서는 Amazon SNS 주제에 대한 구독을 삭제합니다.

```
aws dms delete-event-subscription \
    --subscription-name "my-dms-events"
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 AWS 데이터베이스 마이그레이션 서비스 사용 설명서의 이벤트 및 알림 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteEventSubscription의 섹션을 참조하세요. AWS CLI

### delete-replication-instance

다음 코드 예시에서는 delete-replication-instance을 사용하는 방법을 보여 줍니다.

AWS CLI

복제 인스턴스를 삭제하려면

다음 delete-replication-instance 예시에서는 복제 인스턴스를 삭제합니다.

```
aws dms delete-replication-instance \
    --replication-instance-arn arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE
```

출력:

```
{
    "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:T3OM7OUB5NM2LCVZF7JPGJRNUE",
        "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"
    }
}
```

자세한 내용은 AWS 데이터베이스 마이그레이션 서비스 사용 설명서의 복제 인스턴스 작업을 AWS DMS 참조하세요.

- 자세한 API 내용은 명령 참조DeleteReplicationInstance의 섹션을 참조하세요. AWS CLI

### `delete-replication-subnet-group`

다음 코드 예시에서는 `delete-replication-subnet-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

서브넷 그룹을 삭제하려면

다음 `delete-replication-subnet-group` 예제에서는 서브넷 그룹을 삭제합니다.

```
aws dms delete-replication-subnet-group \
--replication-subnet-group-identifier my-subnet-group
```

출력:

```
(none)
```

자세한 내용은 AWS 데이터베이스 마이그레이션 서비스 사용 설명서의 복제 인스턴스용 네트워크 설정을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteReplicationSubnetGroup의 섹션을 참조하세요. AWS CLI

### delete-replication-task

다음 코드 예시에서는 delete-replication-task을 사용하는 방법을 보여 줍니다.

AWS CLI

복제 작업을 삭제하려면

다음 delete-replication-task 예제에서는 복제 작업을 삭제합니다.

```
aws dms delete-replication-task \
    --replication-task-arn arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII
```

출력:

```
{
    "ReplicationTask": {
        "ReplicationTaskIdentifier": "moveit2",
        "SourceEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
        "TargetEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:EOM4SFKCZEYHZBFGAGZT3QEC5U",
        "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE",
        "MigrationType": "full-load",
        "TableMappings": ...output omitted...,
        "ReplicationTaskSettings": ...output omitted...,
        "Status": "deleting",
        "StopReason": "Stop Reason FULL_LOAD_ONLY_FINISHED",
        "ReplicationTaskCreationDate": 1590524772.505,
        "ReplicationTaskStartDate": 1590789988.677,
        "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"
    }
}
```

자세한 내용은 AWS 데이터베이스 마이그레이션 서비스 사용 설명서의 작업 작업을 AWS DMS 참조하세요.

- 자세한 API 내용은 명령 참조DeleteReplicationTask의 섹션을 참조하세요. AWS CLI

### describe-account-attributes

다음 코드 예시에서는 describe-account-attributes을 사용하는 방법을 보여 줍니다.

AWS CLI

계정 속성을 설명하려면

다음 describe-account-attributes 예제에서는 AWS 계정의 속성을 나열합니다.

```
aws dms describe-account-attributes
```

출력:

```
{
    "AccountQuotas": [
        {
            "AccountQuotaName": "ReplicationInstances",
            "Used": 1,
            "Max": 20
        },
        {
            "AccountQuotaName": "AllocatedStorage",
            "Used": 5,
            "Max": 10000
        },

        ...remaining output omitted...

    ],
    "UniqueAccountIdentifier": "cqahfbfy5xee"
}
```

- 자세한 API 내용은 명령 참조DescribeAccountAttributes의 섹션을 참조하세요. AWS CLI

### describe-certificates

다음 코드 예시에서는 describe-certificates을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 인증서를 나열하려면

다음 describe-certificates 예제에서는 AWS 계정에서 사용 가능한 인증서를 나열합니다.

```
aws dms describe-certificates
```

출력:

```
{
    "Certificates": [
        {
            "CertificateIdentifier": "my-cert",
            "CertificateCreationDate": 1543259542.506,
            "CertificatePem": "-----BEGIN CERTIFICATE-----
\nMIID9DCCAtygAwIBAgIBQjANBgkqhkiG9w0BAQ ...U"

            ... remaining output omittted ...

        }
    ]
}
```

자세한 내용은 AWS 데이터베이스 마이그레이션 서비스 사용 설명서의 사용을 SSL 참조하세요.

- 자세한 API 내용은 명령 참조DescribeCertificates의 섹션을 참조하세요. AWS CLI

### describe-connections

다음 코드 예시에서는 describe-connections을 사용하는 방법을 보여 줍니다.

AWS CLI

연결을 설명하려면

다음 describe-connections 예제에서는 복제 인스턴스와 엔드포인트 간에 테스트한 연결을 나열합니다.

```
aws dms describe-connections
```

출력:

```
{
    "Connections": [
        {
```

```
            "Status": "successful",
            "ReplicationInstanceIdentifier": "test",
            "EndpointArn": "arn:aws:dms:us-east-arn:aws:dms:us-
east-1:123456789012:endpoint:ZW5UAN6P4E77EC7YWHK4RZZ3BE",
            "EndpointIdentifier": "testsrc1",
            "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:6UTDJGBOUS3VI3SUWA66XFJCJQ"
        }
    ]
}
```

자세한 내용은 AWS 데이터베이스 마이그레이션 서비스 사용 설명서의 소스 및 대상 엔드포인트 생성을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeConnections의 섹션을 참조하세요. AWS CLI

### `describe-endpoint-types`

다음 코드 예시에서는 `describe-endpoint-types`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 엔드포인트 유형을 나열하려면

다음 `describe-endpoint-types` 예제에서는 사용 가능한 내SQL 엔드포인트 유형을 나열합니다.

```
aws dms describe-endpoint-types \
    --filters "Name=engine-name,Values=mysql"
```

출력:

```
{
    "SupportedEndpointTypes": [
        {
            "EngineName": "mysql",
            "SupportsCDC": true,
            "EndpointType": "source",
            "EngineDisplayName": "MySQL"
        },
        {
            "EngineName": "mysql",
```

```
            "SupportsCDC": true,
            "EndpointType": "target",
            "EngineDisplayName": "MySQL"
        }
    ]
}
```

자세한 내용은 AWS 데이터베이스 마이그레이션 서비스 사용 설명서의 엔드포인트 작업 AWS DMS <https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Endpoints.html>`__을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeEndpointTypes의 섹션을 참조하세요. AWS CLI

### `describe-endpoints`

다음 코드 예시에서는 `describe-endpoints`을 사용하는 방법을 보여 줍니다.

AWS CLI

엔드포인트를 설명하려면

다음 `describe-endpoints` 예제에서는 AWS 계정의 엔드포인트를 나열합니다.

```
aws dms describe-endpoints
```

출력:

```
{
    "Endpoints": [
        {
            "Username": "dms",
            "Status": "active",
            "EndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:SF2WOFLWYWKVEOHID2EKLP3SJI",
            "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
        }
    ]
}
```

자세한 내용은 AWS 데이터베이스 마이그레이션 서비스 사용 설명서의 엔드포인트 작업을 AWS DMS 참조하세요.

- 자세한 API 내용은 명령 참조DescribeEndpoints의 섹션을 참조하세요. AWS CLI

### `describe-event-categories`

다음 코드 예시에서는 `describe-event-categories`을 사용하는 방법을 보여 줍니다.

AWS CLI

이벤트 범주를 설명하려면

다음 `describe-event-categories` 예제에서는 사용 가능한 이벤트 범주를 나열합니다.

```
aws dms describe-event-categories
```

출력:

```
{
    "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"
            ]
        }
    ]
}
```

자세한 내용은 AWS 데이터베이스 마이그레이션 서비스 사용 설명서의 이벤트 및 알림 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeEventCategories의 섹션을 참조하세요. AWS CLI

### describe-event-subscriptions

다음 코드 예시에서는 describe-event-subscriptions을 사용하는 방법을 보여 줍니다.

AWS CLI

이벤트 구독을 설명하려면

다음 describe-event-subscriptions 예제에서는 Amazon SNS 주제에 대한 이벤트 구독을 나열합니다.

```
aws dms describe-event-subscriptions
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 AWS 데이터베이스 마이그레이션 서비스 사용 설명서의 이벤트 및 알림 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeEventSubscriptions의 섹션을 참조하세요. AWS CLI

### `describe-events`

다음 코드 예시에서는 `describe-events`을 사용하는 방법을 보여 줍니다.

AWS CLI

DMS 이벤트를 나열하려면

다음 `describe-events` 예제에서는 복제 인스턴스에서 시작된 이벤트를 나열합니다.

```
aws dms describe-events \
    --source-type "replication-instance"
```

출력:

```
{
    "Events": [
        {
            "SourceIdentifier": "my-repl-instance",
            "SourceType": "replication-instance",
            "Message": "Replication application shutdown",
            "EventCategories": [],
            "Date": 1590771645.776
        }
    ]
}
```

자세한 내용은 AWS 데이터베이스 마이그레이션 서비스 사용 설명서의 이벤트 및 알림 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeEvents의 섹션을 참조하세요. AWS CLI

### `describe-orderable-replication-instances`

다음 코드 예시에서는 `describe-orderable-replication-instances`을 사용하는 방법을 보여 줍니다.

AWS CLI

주문 가능한 복제 인스턴스를 설명하려면

다음 `describe-orderable-replication-instances` 예제에서는 주문할 수 있는 복제 인스턴스 유형을 나열합니다.

```
aws dms describe-orderable-replication-instances
```

출력:

```
{
    "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...

    }
```

자세한 내용은 AWS 데이터베이스 마이그레이션 서비스 사용 설명서의 복제 인스턴스 작업을 AWS DMS 참조하세요.

- 자세한 API 내용은 명령 참조DescribeOrderableReplicationInstances의 섹션을 참조하세요. AWS CLI

### describe-refresh-schemas-status

다음 코드 예시에서는 describe-refresh-schemas-status을 사용하는 방법을 보여 줍니다.

AWS CLI

엔드포인트의 새로 고침 상태를 나열하려면

다음 describe-refresh-schemas-status 예제에서는 이전 새로 고침 요청의 상태를 반환합니다.

```
aws dms describe-refresh-schemas-status \
    --endpoint-arn arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA
```

출력:

```
{
```

```
    "RefreshSchemasStatus": {
        "EndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
        "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE",
        "Status": "successful",
        "LastRefreshDate": 1590786544.605
    }
}
```

- 자세한 API 내용은 명령 참조DescribeRefreshSchemasStatus의 섹션을 참조하세요. AWS CLI

### describe-replication-instances

다음 코드 예시에서는 describe-replication-instances을 사용하는 방법을 보여 줍니다.

AWS CLI

복제 인스턴스를 설명하려면

다음 describe-replication-instances 예제에서는 AWS 계정의 복제 인스턴스를 나열합니다.

```
aws dms describe-replication-instances
```

출력:

```
{
    "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:T3OM7OUB5NM2LCVZF7JPGJRNUE",
            "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
        }
    ]
}
```

자세한 내용은 AWS 데이터베이스 마이그레이션 서비스 사용 설명서의 복제 인스턴스 작업을 AWS DMS 참조하세요.

- 자세한 API 내용은 명령 참조DescribeReplicationInstances의 섹션을 참조하세요. AWS CLI

### `describe-replication-subnet-groups`

다음 코드 예시에서는 `describe-replication-subnet-groups`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 서브넷 그룹을 표시하려면

다음 describe-replication-subnet-groups 예제에서는 사용 가능한 서브넷 그룹을 나열합니다.

```
aws dms describe-replication-subnet-groups \
    --filter "Name=replication-subnet-group-id,Values=my-subnet-group"
```

출력:

```
{
    "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"
                }
            ]
        }
    ]
}
```

자세한 내용은 AWS 데이터베이스 마이그레이션 서비스 사용 설명서의 복제 인스턴스에 대한 네트워크 설정을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeReplicationSubnetGroups의 섹션을 참조하세요. AWS CLI

### describe-replication-task-assessment-results

다음 코드 예시에서는 describe-replication-task-assessment-results을 사용하는 방법을 보여 줍니다.

AWS CLI

복제 작업 평가 결과를 나열하려면

다음 describe-replication-task-assessment-results 예제에서는 이전 작업 평가의 결과를 나열합니다.

```
aws dms describe-replication-task-assessment-results
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS 데이터베이스 마이그레이션 서비스 사용 설명서의 작업 평가 보고서 생성을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeReplicationTaskAssessmentResults의 섹션을 참조하세요. AWS CLI

### describe-replication-tasks

다음 코드 예시에서는 describe-replication-tasks을 사용하는 방법을 보여 줍니다.

AWS CLI

복제 작업을 설명하려면

다음 describe-replication-tasks 예제에서는 현재 복제 작업을 설명합니다.

```
aws dms describe-replication-tasks
```

출력:

```
{
    "ReplicationTasks": [
        {
            "ReplicationTaskIdentifier": "moveit2",
            "SourceEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
            "TargetEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:EOM4SFKCZEYHZBFGAGZT3QEC5U",
            "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE",
            "MigrationType": "full-load",
            "TableMappings": ...output omitted... ,
            "ReplicationTaskSettings": ...output omitted... ,
            "Status": "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
```

```
            }
        }
    ]
}
```

자세한 내용은 AWS 데이터베이스 마이그레이션 서비스 사용 설명서의 작업 작업을 AWS DMS 참조하세요.

- 자세한 API 내용은 명령 참조DescribeReplicationTasks의 섹션을 참조하세요. AWS CLI

### describe-schemas

다음 코드 예시에서는 describe-schemas을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터베이스 스키마를 설명하려면

다음 describe-schemas 예제에서는 엔드포인트에서 사용 가능한 테이블을 나열합니다.

```
aws dms describe-schemas \
    --endpoint-arn "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA"
```

출력:

```
{
    "Schemas": [
        "prodrep"
    ]
}
```

자세한 내용은 AWS 데이터베이스 마이그레이션 서비스 사용 설명서의 주제 제목을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeSchemas의 섹션을 참조하세요. AWS CLI

### list-tags-for-resource

다음 코드 예시에서는 list-tags-for-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스의 태그를 나열하려면

다음 list-tags-for-resource 예제에서는 복제 인스턴스의 태그를 나열합니다.

```
aws dms list-tags-for-resource \
    --resource-arn arn:aws:dms:us-east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE
```

출력:

```
{
    "TagList": [
        {
            "Key": "Project",
            "Value": "dbMigration"
        },
        {
            "Key": "Environment",
            "Value": "PROD"
        }
    ]
}
```

자세한 내용은 AWS 데이터베이스 마이그레이션 서비스 사용 설명서의 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### modify-endpoint

다음 코드 예시에서는 modify-endpoint을 사용하는 방법을 보여 줍니다.

AWS CLI

엔드포인트를 수정하려면

다음 modify-endpoint 예제에서는 엔드포인트에 추가 연결 속성을 추가합니다.

```
aws dms modify-endpoint \
    --endpoint-arn "arn:aws:dms:us-
east-1:123456789012:endpoint:GUVAFG34EECUOJ6QVZ56DAHT3U" \
    --extra-connection-attributes "compressionType=GZIP"
```

출력:

```
{
    "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:GUVAFG34EECUOJ6QVZ56DAHT3U",
        "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
        }
    }
}
```

자세한 내용은 AWS 데이터베이스 마이그레이션 서비스 사용 설명서의 엔드포인트 작업 AWS DMS <https://docs.aws.amazon.com/dms/latest/userguide/CHAP_Endpoints.html>`__을 참조하세요.

- 자세한 API 내용은 명령 참조ModifyEndpoint의 섹션을 참조하세요. AWS CLI

### `modify-event-subscription`

다음 코드 예시에서는 `modify-event-subscription`을 사용하는 방법을 보여 줍니다.

AWS CLI

이벤트 구독을 수정하려면

다음 `modify-event-subscription` 예제에서는 이벤트 구독의 소스 유형을 변경합니다.

```
aws dms modify-event-subscription \
```

```
    --subscription-name "my-dms-events" \
    --source-type replication-task
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 AWS 데이터베이스 마이그레이션 서비스 사용 설명서의 이벤트 및 알림 작업을 참조하세요.

- 자세한 API 내용은 명령 참조ModifyEventSubscription의 섹션을 참조하세요. AWS CLI

### `modify-replication-instance`

다음 코드 예시에서는 `modify-replication-instance`을 사용하는 방법을 보여 줍니다.

AWS CLI

복제 인스턴스를 수정하려면

다음 `modify-replication-instance` 예제에서는 다중 AZ 배포를 사용하도록 복제 인스턴스를 수정합니다.

```
aws dms modify-replication-instance \
     --replication-instance-arn arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE \
     --multi-az
```

출력:

```
{
```

```
    "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...

    }
}
```

자세한 내용은 AWS 데이터베이스 마이그레이션 서비스 사용 설명서의 복제 인스턴스 작업을 AWS DMS 참조하세요.

- 자세한 API 내용은 명령 참조ModifyReplicationInstance의 섹션을 참조하세요. AWS CLI

### `modify-replication-subnet-group`

다음 코드 예시에서는 `modify-replication-subnet-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

서브넷 그룹을 수정하려면

다음 `modify-replication-subnet-group` 예제에서는 서브넷 그룹과 연결된 서브넷 목록을 변경합니다.

```
aws dms modify-replication-subnet-group \
    --replication-subnet-group-identifier my-subnet-group \
    --subnet-id subnet-da327bf6 subnet-bac383e0
```

출력:

```
{
    "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"
            }
        ]
    }
}
```

자세한 내용은 AWS 데이터베이스 마이그레이션 서비스 사용 설명서의 복제 인스턴스에 대한 네트워크 설정을 참조하세요.

- 자세한 API 내용은 명령 참조ModifyReplicationSubnetGroup의 섹션을 참조하세요. AWS CLI

### `modify-replication-task`

다음 코드 예시에서는 `modify-replication-task`을 사용하는 방법을 보여 줍니다.

AWS CLI

복제 작업을 수정하려면

다음 `modify-replication-task` 예제에서는 태스크의 테이블 매핑을 변경합니다.

```
aws dms modify-replication-task \
```

```
    --replication-task-arn "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII" \
    --table-mappings file://table-mappings.json
```

table-mappings.json의 콘텐츠:

```
{
    "rules": [
        {
            "rule-type": "selection",
            "rule-id": "1",
            "rule-name": "1",
            "object-locator": {
                "schema-name": "prodrep",
                "table-name": "ACCT_%"
            },
            "rule-action": "include",
            "filters": []
        }
    ]
}
```

출력:

```
{
    "ReplicationTask": {
        "ReplicationTaskIdentifier": "moveit2",
        "SourceEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
        "TargetEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:EOM4SFKCZEYHZBFGAGZT3QEC5U",
        "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE",
        "MigrationType": "full-load",
        "TableMappings": ...output omitted...,
        "ReplicationTaskSettings": ...output omitted...,
        "Status": "modifying",
        "StopReason": "Stop Reason FULL_LOAD_ONLY_FINISHED",
        "ReplicationTaskCreationDate": 1590524772.505,
        "ReplicationTaskStartDate": 1590789424.653,
        "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"
    }
```

```
}
```

자세한 내용은 AWS 데이터베이스 마이그레이션 서비스 사용 설명서의 작업 작업을 AWS DMS 참조하세요.

- 자세한 API 내용은 명령 참조ModifyReplicationTask의 섹션을 참조하세요. AWS CLI

### reboot-replication-instance

다음 코드 예시에서는 reboot-replication-instance을 사용하는 방법을 보여 줍니다.

AWS CLI

복제 인스턴스를 재부팅하려면

다음 reboot-replication-instance 예제에서는 복제 인스턴스를 재부팅합니다.

```
aws dms reboot-replication-instance \
    --replication-instance-arn arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE
```

출력:

```
{
    "ReplicationInstance": {
        "ReplicationInstanceIdentifier": "my-repl-instance",
        "ReplicationInstanceClass": "dms.t2.micro",
        "ReplicationInstanceStatus": "rebooting",
        "AllocatedStorage": 5,
        "InstanceCreateTime": 1590011235.952,
    ... output omitted ...
    }
}
```

자세한 내용은 AWS 데이터베이스 마이그레이션 서비스 사용 설명서의 복제 인스턴스 작업을 AWS DMS 참조하세요.

- 자세한 API 내용은 명령 참조RebootReplicationInstance의 섹션을 참조하세요. AWS CLI

### refresh-schemas

다음 코드 예시에서는 refresh-schemas을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터베이스 스키마를 새로 고치려면

다음 `refresh-schemas` 예제는 엔드포인트에서 스키마 목록을 새로 AWS DMS고치는 요청입니다.

```
aws dms refresh-schemas \
    --replication-instance-arn arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE \
    --endpoint-arn "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA"
```

출력:

```
{
    "RefreshSchemasStatus": {
        "EndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
        "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE",
        "Status": "refreshing",
        "LastRefreshDate": 1590019949.103
    }
}
```

- 자세한 API 내용은 명령 참조RefreshSchemas의 섹션을 참조하세요. AWS CLI

### `reload-tables`

다음 코드 예시에서는 `reload-tables`을 사용하는 방법을 보여 줍니다.

AWS CLI

엔드포인트에서 사용할 수 있는 테이블 목록을 새로 고치려면

다음 `reload-tables` 예제에서는 엔드포인트에서 사용 가능한 테이블 목록을 다시 로드합니다.

```
aws dms reload-tables \
    --replication-task-arn "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII" \
    --tables-to-reload "SchemaName=prodrep,TableName=ACCT_BAL"
```

출력:

```
{
    "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"
}
```

- 자세한 API 내용은 명령 참조ReloadTables의 섹션을 참조하세요. AWS CLI

### `remove-tags-from-resource`

다음 코드 예시에서는 `remove-tags-from-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

복제 인스턴스에서 태그를 제거하려면

다음 `remove-tags-from-resource` 예제에서는 복제 인스턴스에서 태그를 제거합니다.

```
aws dms remove-tags-from-resource \
    --resource-arn arn:aws:dms:us-east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE
 \
    --tag-keys Environment Project
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS 데이터베이스 마이그레이션 서비스 사용 설명서의 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조RemoveTagsFromResource의 섹션을 참조하세요. AWS CLI

### `start-replication-task-assessment`

다음 코드 예시에서는 `start-replication-task-assessment`을 사용하는 방법을 보여 줍니다.

AWS CLI

작업 평가를 시작하려면

다음 `start-replication-task-assessment` 예제에서는 복제 작업 평가를 시작합니다.

```
aws dms start-replication-task-assessment \
```

```
    --replication-task-arn arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII
```

출력:

```
{
    "ReplicationTask": {
        "ReplicationTaskIdentifier": "moveit2",
        "SourceEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
        "TargetEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:EOM4SFKCZEYHZBFGAGZT3QEC5U",
        "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE",
        "MigrationType": "full-load",
        "TableMappings": ...output omitted...,
        "ReplicationTaskSettings": ...output omitted...,
        "Status": "testing",
        "StopReason": "Stop Reason FULL_LOAD_ONLY_FINISHED",
        "ReplicationTaskCreationDate": 1590524772.505,
        "ReplicationTaskStartDate": 1590789988.677,
        "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"
    }
}
```

자세한 내용은 AWS 데이터베이스 마이그레이션 서비스 사용 설명서의 작업 평가 보고서 생성을 참조하세요.

- 자세한 API 내용은 명령 참조StartReplicationTaskAssessment의 섹션을 참조하세요. AWS CLI

### `start-replication-task`

다음 코드 예시에서는 `start-replication-task`을 사용하는 방법을 보여 줍니다.

AWS CLI

복제 작업을 시작하려면

다음 `command-name` 예제에서는 AWS 계정에서 사용 가능한 위젯을 나열합니다.

```
aws dms start-replication-task \
```

```
    --replication-task-arn arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII \
    --start-replication-task-type reload-target
```

출력:

```
{
    "ReplicationTask": {
        "ReplicationTaskIdentifier": "moveit2",
        "SourceEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
        "TargetEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:EOM4SFKCZEYHZBFGAGZT3QEC5U",
        "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE",
        "MigrationType": "full-load",
        "TableMappings": ...output omitted... ,
        "ReplicationTaskSettings": ...output omitted... ,
        "Status": "starting",
        "ReplicationTaskCreationDate": 1590524772.505,
        "ReplicationTaskStartDate": 1590619805.212,
        "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"
    }
}
```

자세한 내용은 AWS 데이터베이스 마이그레이션 서비스 사용 설명서의 작업 작업을 AWS DMS 참조하세요.

- 자세한 API 내용은 명령 참조StartReplicationTask의 섹션을 참조하세요. AWS CLI

### stop-replication-task

다음 코드 예시에서는 stop-replication-task을 사용하는 방법을 보여 줍니다.

AWS CLI

작업을 중지하려면

다음 stop-replication-task 예제는 태스크를 중지합니다.

```
aws dms stop-replication-task \
```

```
    --replication-task-arn arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII
```

출력:

```
{
    "ReplicationTask": {
        "ReplicationTaskIdentifier": "moveit2",
        "SourceEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
        "TargetEndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:EOM4SFKCZEYHZBFGAGZT3QEC5U",
        "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE",
        "MigrationType": "full-load",
        "TableMappings": ...output omitted...,
        "ReplicationTaskSettings": ...output omitted...,
        "Status": "stopping",
        "ReplicationTaskCreationDate": 1590524772.505,
        "ReplicationTaskStartDate": 1590789424.653,
        "ReplicationTaskArn": "arn:aws:dms:us-
east-1:123456789012:task:K55IUCGBASJS5VHZJIINA45FII"
    }
}
```

자세한 내용은 AWS 데이터베이스 마이그레이션 서비스 사용 설명서의 작업 작업을 AWS DMS 참조하세요.

- 자세한 API 내용은 명령 참조StopReplicationTask의 섹션을 참조하세요. AWS CLI

### `test-connection`

다음 코드 예시에서는 `test-connection`을 사용하는 방법을 보여 줍니다.

AWS CLI

엔드포인트에 대한 연결을 테스트하려면

다음 `test-connection` 예제에서는 복제 인스턴스에서 엔드포인트에 액세스할 수 있는지 테스트합니다.

```
aws dms test-connection \
```

```
    --replication-instance-arn arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE \
    --endpoint-arn arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA
```

출력:

```
{
    "Connection": {
        "ReplicationInstanceArn": "arn:aws:dms:us-
east-1:123456789012:rep:T3OM7OUB5NM2LCVZF7JPGJRNUE",
        "EndpointArn": "arn:aws:dms:us-
east-1:123456789012:endpoint:6GGI6YPWWGAYUVLKIB732KEVWA",
        "Status": "testing",
        "EndpointIdentifier": "src-database-1",
        "ReplicationInstanceIdentifier": "my-repl-instance"
    }
}
```

자세한 내용은 AWS 데이터베이스 마이그레이션 서비스 사용 설명서의 소스 및 대상 엔드포인트 생성을 참조하세요.

- 자세한 API 내용은 명령 참조TestConnection의 섹션을 참조하세요. AWS CLI

## 를 사용한 Amazon DocumentDB 예제 AWS CLI

다음 코드 예제에서는 Amazon DocumentDB 와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### add-tags-to-resource

다음 코드 예시에서는 add-tags-to-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 리소스에 하나 이상의 태그를 추가하려면

다음 add-tags-to-resource 예제에서는 에 세 개의 태그를 추가합니다sample-cluster. 하나의 태그(CropB)에 키 이름이 있지만 값이 없습니다.

```
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"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon DocumentDB 개발자 안내서의 Amazon DocumentDB 리소스 태그 지정을 참조하세요. Amazon DocumentDB

- 자세한 API 내용은 명령 참조AddTagsToResource의 섹션을 참조하세요. AWS CLI

### apply-pending-maintenance-action

다음 코드 예시에서는 apply-pending-maintenance-action을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 유지 관리 기간 동안 보류 중인 유지 관리 작업이 수행되도록 하려면

다음 apply-pending-maintenance-action 예제에서는 예약된 다음 유지 관리 기간 동안 모든 시스템 업데이트 작업을 수행합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon DocumentDB 개발자 안내서의 Amazon DocumentDB 업데이트 적용을 참조하세요. Amazon DocumentDB

- 자세한 API 내용은 명령 참조ApplyPendingMaintenanceAction의 섹션을 참조하세요. AWS CLI

### copy-db-cluster-parameter-group

다음 코드 예시에서는 copy-db-cluster-parameter-group을 사용하는 방법을 보여 줍니다.

AWS CLI

기존 DB 클러스터 파라미터 그룹을 복제하려면

다음 copy-db-cluster-parameter-group 예제에서는 custom-docdb3-6라는 파라미터 그룹의 복사본을 만듭니다custom-docdb3-6-copy. 복사본을 만들 때 새 파라미터 그룹에 태그를 추가합니다.

```
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"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon DocumentDB 개발자 안내서의 Amazon DocumentDB 클러스터 파라미터 그룹 복사를 참조하세요. Amazon DocumentDB

- 자세한 API 내용은 명령 참조CopyDbClusterParameterGroup의 섹션을 참조하세요. AWS CLI

### copy-db-cluster-snapshot

다음 코드 예시에서는 copy-db-cluster-snapshot을 사용하는 방법을 보여 줍니다.

AWS CLI

스냅샷 사본을 생성하려면

다음 copy-db-cluster-snapshot 예에서는 sample-cluster-snapshot-copy라는 이름으로 sample-cluster-snapshot의 복사본을 생성합니다. 복사본에는 원본의 모든 태그와 키 이름이 인 새 태그가 있습니다CopyNumber.

```
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"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon DocumentDB 개발자 안내서의 클러스터 스냅샷 복사를 참조하세요.

- 자세한 API 내용은 명령 참조CopyDbClusterSnapshot의 섹션을 참조하세요. AWS CLI

### create-db-cluster-parameter-group

다음 코드 예시에서는 create-db-cluster-parameter-group을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon DocumentDB 클러스터 파라미터 그룹을 생성하려면

다음 create-db-cluster-parameter-group 예제에서는 docdb3.6 패밀리를 sample-parameter-group 사용하여 DB 클러스터 파라미터 그룹을 생성합니다.

```
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"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon DocumentDB 개발자 안내서의 Amazon DocumentDB 클러스터 파라미터 그룹 생성을 참조하세요. Amazon DocumentDB

- 자세한 API 내용은 명령 참조CreateDbClusterParameterGroup의 섹션을 참조하세요. AWS CLI

### `create-db-cluster-snapshot`

다음 코드 예시에서는 `create-db-cluster-snapshot`을 사용하는 방법을 보여 줍니다.

AWS CLI

수동 Amazon DocumentDB 클러스터 스냅샷을 생성하려면

다음 `create-db-cluster-snapshot` 예제에서는 라는 Amazon DB 클러스터 스냅샷을 생성합니다 sample-cluster-snapshot.

```
aws docdb create-db-cluster-snapshot \
   --db-cluster-identifier sample-cluster \
   --db-cluster-snapshot-identifier sample-cluster-snapshot
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon DocumentDB 개발자 안내서의 수동 클러스터 스냅샷 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateDbClusterSnapshot의 섹션을 참조하세요. AWS CLI

### `create-db-cluster`

다음 코드 예시에서는 `create-db-cluster`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon DocumentDB 클러스터를 생성하려면

다음 `create-db-cluster` 예제에서는 일요일 20:30~11:00 사이에 기본 유지 관리 기간`sample-cluster`으로 라는 Amazon DocumentDB 클러스터를 생성합니다.

```
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
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon DocumentDB 개발자 안내서의 Amazon DocumentDB 클러스터 생성을 참조하세요. Amazon DocumentDB

- 자세한 API 내용은 명령 참조CreateDbCluster의 섹션을 참조하세요. AWS CLI

### create-db-instance

다음 코드 예시에서는 create-db-instance을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon DocumentDB 클러스터 인스턴스를 생성하려면

다음 create-db-instance 예제 코드는 Amazon DocumentDB 클러스터 sample-cluster-instance-2 에 인스턴스를 생성합니다sample-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
```

출력:

```
{
    "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-XEKJLEMGRV5ZKCARUVA4HO3ITE"
    }
```

```
}
```

자세한 내용은 Amazon DocumentDB 개발자 안내서의 클러스터에 Amazon DocumentDB 인스턴스 추가를 참조하세요. Amazon DocumentDB

- 자세한 API 내용은 명령 참조CreateDbInstance의 섹션을 참조하세요. AWS CLI

### `create-db-subnet-group`

다음 코드 예시에서는 `create-db-subnet-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon DocumentDB 서브넷 그룹을 생성하려면

다음 `create-db-subnet-group` 예제에서는 라는 Amazon DocumentDB 서브넷 그룹을 생성합니다`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"
```

출력:

```
{
    "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"
                }
            }
        ]
    }
}
```

자세한 내용은 Amazon DocumentDB 개발자 안내서의 Amazon DocumentDB 서브넷 그룹 생성을 참조하세요. Amazon DocumentDB

- 자세한 API 내용은 명령 참조CreateDbSubnetGroup의 섹션을 참조하세요. AWS CLI

### `delete-db-cluster-parameter-group`

다음 코드 예시에서는 `delete-db-cluster-parameter-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon DocumentDB 클러스터 파라미터 그룹을 삭제하려면

다음 `delete-db-cluster-parameter-group` 예제에서는 Amazon DocumentDB 파라미터 그룹 를 삭제합니다`sample-parameter-group`.

```
aws docdb delete-db-cluster-parameter-group \
    --db-cluster-parameter-group-name sample-parameter-group
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon DocumentDB 개발자 안내서의 Amazon DocumentDB 클러스터 파라미터 그룹 삭제를 참조하세요. Amazon DocumentDB

- 자세한 API 내용은 명령 참조DeleteDbClusterParameterGroup의 섹션을 참조하세요. AWS CLI

### `delete-db-cluster-snapshot`

다음 코드 예시에서는 `delete-db-cluster-snapshot`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon DocumentDB 클러스터 스냅샷을 삭제하려면

다음 `delete-db-cluster-snapshot` 예제에서는 Amazon DocumentDB 클러스터 스냅샷 을 삭제합니다`sample-cluster-snapshot`.

```
aws docdb delete-db-cluster-snapshot \
    --db-cluster-snapshot-identifier sample-cluster-snapshot
```

출력:

```
{
    "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
    }
```

```
}
```

자세한 내용은 Amazon DocumentDB 개발자 안내서의 클러스터 스냅샷 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteDbClusterSnapshot의 섹션을 참조하세요. AWS CLI

### `delete-db-cluster`

다음 코드 예시에서는 `delete-db-cluster`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon DocumentDB 클러스터를 삭제하려면

다음 `delete-db-cluster` 예제에서는 Amazon DocumentDB 클러스터 를 삭제합니다`sample-cluster`. 클러스터를 삭제하기 전에는 클러스터를 백업하지 않습니다. NOTE: 클러스터와 연결된 모든 인스턴스를 삭제해야 클러스터를 삭제할 수 있습니다.

```
aws docdb delete-db-cluster \
    --db-cluster-identifier sample-cluster \
    --skip-final-snapshot
```

출력:

```
{
    "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": []
    }
}
```

자세한 내용은 Amazon DocumentDB 개발자 안내서의 Amazon DocumentDB 클러스터 삭제를 참조하세요. Amazon DocumentDB

- 자세한 API 내용은 명령 참조DeleteDbCluster의 섹션을 참조하세요. AWS CLI

### `delete-db-instance`

다음 코드 예시에서는 `delete-db-instance`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon DocumentDB 인스턴스를 삭제하려면

다음 `delete-db-instance` 예제에서는 Amazon DocumentDB 인스턴스 를 삭제합니다`sample-cluster-instance-2`.

```
aws docdb delete-db-instance \
    --db-instance-identifier sample-cluster-instance-2
```

출력:


```
{
    "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-XEKJLEMGRV5ZKCARUVA4HO3ITE",
```

```
        "BackupRetentionPeriod": 3,
        "Engine": "docdb",
        "VpcSecurityGroups": [
            {
                "Status": "active",
                "VpcSecurityGroupId": "sg-77186e0d"
            }
        ],
        "AutoMinorVersionUpgrade": true,
        "PromotionTier": 1,
        "EngineVersion": "3.6.0",
        "Endpoint": {
            "Address": "sample-cluster-instance-2.corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
            "HostedZoneId": "ZNKXH85TT8WVW",
            "Port": 27017
        },
        "DBInstanceIdentifier": "sample-cluster-instance-2",
        "PreferredMaintenanceWindow": "tue:10:28-tue:10:58",
        "EnabledCloudwatchLogsExports": [
            "audit"
        ],
        "PendingModifiedValues": {},
        "DBInstanceStatus": "deleting",
        "PubliclyAccessible": false,
        "DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster-
instance-2",
        "DBClusterIdentifier": "sample-cluster",
        "AvailabilityZone": "us-west-2c",
        "StorageEncrypted": false
    }
}
```

자세한 내용은 Amazon DocumentDB 개발자 안내서의 Amazon DocumentDB 인스턴스 삭제를 참조하세요. Amazon DocumentDB

- 자세한 API 내용은 명령 참조DeleteDbInstance의 섹션을 참조하세요. AWS CLI

### `delete-db-subnet-group`

다음 코드 예시에서는 `delete-db-subnet-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon DocumentDB 서브넷 그룹을 삭제하려면

다음 delete-db-subnet-group 예제에서는 Amazon DocumentDB 서브넷 그룹 를 삭제합니다sample-subnet-group.

```
aws docdb delete-db-subnet-group \
    --db-subnet-group-name sample-subnet-group
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon DocumentDB 개발자 안내서의 Amazon DocumentDB 서브넷 그룹 삭제를 참조하세요. Amazon DocumentDB

- 자세한 API 내용은 명령 참조DeleteDbSubnetGroup의 섹션을 참조하세요. AWS CLI

### describe-db-cluster-parameter-groups

다음 코드 예시에서는 describe-db-cluster-parameter-groups을 사용하는 방법을 보여 줍니다.

AWS CLI

하나 이상의 Amazon DocumentDB 클러스터 파라미터 그룹의 세부 정보를 보려면

다음 describe-db-cluster-parameter-groups 예제에서는 Amazon DocumentDB 클러스터 파라미터 그룹 에 대한 세부 정보를 표시합니다custom3-6-param-grp.

```
aws docdb describe-db-cluster-parameter-groups \
    --db-cluster-parameter-group-name custom3-6-param-grp
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon DocumentDB 개발자 안내서의 Amazon DocumentDB 클러스터 파라미터 그룹 보기를 참조하세요. Amazon DocumentDB

- 자세한 API 내용은 명령 참조DescribeDbClusterParameterGroups의 섹션을 참조하세요. AWS CLI

### `describe-db-cluster-parameters`

다음 코드 예시에서는 `describe-db-cluster-parameters`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon DocumentDB 클러스터 파라미터 그룹의 세부 파라미터 목록을 봅니다.

다음 `describe-db-cluster-parameters` 예제에서는 Amazon DocumentDB 파라미터 그룹 custom3-6-param-grp의 파라미터를 나열합니다.

```
aws docdb describe-db-cluster-parameters \
     --db-cluster-parameter-group-name custom3-6-param-grp
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon DocumentDB 개발자 안내서의 Amazon DocumentDB 클러스터 파라미터 보기를 참조하세요. Amazon DocumentDB

- 자세한 API 내용은 명령 참조DescribeDbClusterParameters의 섹션을 참조하세요. AWS CLI

### `describe-db-cluster-snapshot-attributes`

다음 코드 예시에서는 `describe-db-cluster-snapshot-attributes`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon DocumentDB 스냅샷 속성 이름 및 값을 나열하려면

다음 `describe-db-cluster-snapshot-attributes` 예제에서는 Amazon DocumentDB 스냅샷 의 속성 이름과 값을 나열합니다`sample-cluster-snapshot`.

```
aws docdb describe-db-cluster-snapshot-attributes \
```

```
    --db-cluster-snapshot-identifier sample-cluster-snapshot
```

출력:

```
{
    "DBClusterSnapshotAttributesResult": {
        "DBClusterSnapshotAttributes": [
            {
                "AttributeName": "restore",
                "AttributeValues": []
            }
        ],
        "DBClusterSnapshotIdentifier": "sample-cluster-snapshot"
    }
}
```

자세한 내용은 Amazon DocumentDB 개발자 안내서의 DescribeDBClusterSnapshotAttributes를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeDbClusterSnapshotAttributes의 섹션을 참조하세요. AWS CLI

### describe-db-cluster-snapshots

다음 코드 예시에서는 describe-db-cluster-snapshots을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon DocumentDB 스냅샷을 설명하려면

다음 describe-db-cluster-snapshots 예제에서는 Amazon DocumentDB 스냅샷 에 대한 세부 정보를 표시합니다sample-cluster-snapshot.

```
aws docdb describe-db-cluster-snapshots \
    --db-cluster-snapshot-identifier sample-cluster-snapshot
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon DocumentDB 개발자 안내서의 D escribeDBCluster스냅샷을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeDbClusterSnapshots의 섹션을 참조하세요. AWS CLI

### `describe-db-clusters`

다음 코드 예시에서는 `describe-db-clusters`을 사용하는 방법을 보여 줍니다.

AWS CLI

하나 이상의 Amazon DocumentDB 클러스터에 대한 자세한 정보를 가져옵니다.

다음 `describe-db-clusters` 예제에서는 Amazon DocumentDB 클러스터 에 대한 세부 정보를 표시합니다`sample-cluster`. `--db-cluster-identifier` 파라미터를 생략하면 최대 100개의 클러스터에 대한 정보를 얻을 수 있습니다.

```
aws docdb describe-db-clusters
    --db-cluster-identifier sample-cluster
```

출력:

```
{
    "DBClusters": [
        {
            "DBClusterParameterGroup": "default.docdb3.6",
            "Endpoint": "sample-cluster.cluster-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
            "PreferredBackupWindow": "00:00-00:30",
            "DBClusterIdentifier": "sample-cluster",
            "ClusterCreateTime": "2019-03-15T20:29:58.836Z",
            "LatestRestorableTime": "2019-03-18T20:28:03.239Z",
            "MasterUsername": "master-user",
            "DBClusterMembers": [
                {
                    "PromotionTier": 1,
                    "DBClusterParameterGroupStatus": "in-sync",
                    "IsClusterWriter": false,
                    "DBInstanceIdentifier": "sample-cluster"
                },
                {
                    "PromotionTier": 1,
                    "DBClusterParameterGroupStatus": "in-sync",
                    "IsClusterWriter": true,
                    "DBInstanceIdentifier": "sample-cluster2"
                }
            ],
            "PreferredMaintenanceWindow": "sat:04:30-sat:05:00",
            "VpcSecurityGroups": [
                {
                    "VpcSecurityGroupId": "sg-77186e0d",
                    "Status": "active"
                }
            ],
            "Engine": "docdb",
            "ReaderEndpoint": "sample-cluster.cluster-ro-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
            "DBSubnetGroup": "default",
            "MultiAZ": true,
            "AvailabilityZones": [
                "us-west-2a",
                "us-west-2c",
                "us-west-2b"
            ],
```

```
            "EarliestRestorableTime": "2019-03-15T20:30:47.020Z",
            "DbClusterResourceId": "cluster-UP4EF2PVDDFVHHDJQTYDAIGHLE",
            "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-
cluster",
            "BackupRetentionPeriod": 3,
            "HostedZoneId": "ZNKXH85TT8WVW",
            "StorageEncrypted": false,
            "EnabledCloudwatchLogsExports": [
                "audit"
            ],
            "AssociatedRoles": [],
            "EngineVersion": "3.6.0",
            "Port": 27017,
            "Status": "available"
        }
    ]
}
```

자세한 내용은 Amazon DocumentDB 개발자 안내서의 Amazon DocumentDB 클러스터 설명을 참조하세요. Amazon DocumentDB

- 자세한 API 내용은 명령 참조DescribeDbClusters의 섹션을 참조하세요. AWS CLI

### `describe-db-engine-versions`

다음 코드 예시에서는 `describe-db-engine-versions`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 Amazon DocumentDB 엔진 버전을 나열하려면

다음 `describe-db-engine-versions` 예제에서는 사용 가능한 모든 Amazon DocumentDB 엔진 버전을 나열합니다.

```
aws docdb describe-db-engine-versions \
    --engine docdb
```

출력:

```
{
    "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"
            ]
        }
    ]
}
```

자세한 내용은 Amazon DocumentDB 개발자 안내서의 D escribeDBEngine버전을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeDbEngineVersions의 섹션을 참조하세요. AWS CLI

### `describe-db-instances`

다음 코드 예시에서는 `describe-db-instances`을 사용하는 방법을 보여 줍니다.

AWS CLI

프로비저닝된 Amazon DocumentDB 인스턴스에 대한 정보를 찾으려면

다음 `describe-db-instances` 예제에서는 Amazon DocumentDB 인스턴스 에 대한 의 세부 정보를 보여줍니다`sample-cluster-instance`. `--db-instance-identifier` 파라미터를 생략하면 최대 100개의 인스턴스에 대한 정보를 얻을 수 있습니다.

```
aws docdb describe-db-instances \
    --db-instance-identifier sample-cluster-instance
```

출력:

```
{
    "DBInstances": [
        {
            "Endpoint": {
                "HostedZoneId": "ZNKXH85TT8WVW",
                "Address": "sample-cluster-instance.corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
                "Port": 27017
            },
```

```
            "PreferredBackupWindow": "00:00-00:30",
            "DBInstanceStatus": "available",
            "DBInstanceClass": "db.r4.large",
            "EnabledCloudwatchLogsExports": [
                "audit"
            ],
            "DBInstanceIdentifier": "sample-cluster-instance",
            "DBSubnetGroup": {
                "Subnets": [
                    {
                        "SubnetStatus": "Active",
                        "SubnetIdentifier": "subnet-4e26d263",
                        "SubnetAvailabilityZone": {
                            "Name": "us-west-2a"
                        }
                    },
                    {
                        "SubnetStatus": "Active",
                        "SubnetIdentifier": "subnet-afc329f4",
                        "SubnetAvailabilityZone": {
                            "Name": "us-west-2c"
                        }
                    },
                    {
                        "SubnetStatus": "Active",
                        "SubnetIdentifier": "subnet-53ab3636",
                        "SubnetAvailabilityZone": {
                            "Name": "us-west-2d"
                        }
                    },
                    {
                        "SubnetStatus": "Active",
                        "SubnetIdentifier": "subnet-991cb8d0",
                        "SubnetAvailabilityZone": {
                            "Name": "us-west-2b"
                        }
                    }
                ],
                "DBSubnetGroupName": "default",
                "SubnetGroupStatus": "Complete",
                "DBSubnetGroupDescription": "default",
                "VpcId": "vpc-91280df6"
            },
            "InstanceCreateTime": "2019-03-15T20:36:06.338Z",
```

```
            "Engine": "docdb",
            "StorageEncrypted": false,
            "AutoMinorVersionUpgrade": true,
            "DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster-
instance",
            "PreferredMaintenanceWindow": "tue:08:39-tue:09:09",
            "VpcSecurityGroups": [
                {
                    "Status": "active",
                    "VpcSecurityGroupId": "sg-77186e0d"
                }
            ],
            "DBClusterIdentifier": "sample-cluster",
            "PendingModifiedValues": {},
            "BackupRetentionPeriod": 3,
            "PubliclyAccessible": false,
            "EngineVersion": "3.6.0",
            "PromotionTier": 1,
            "AvailabilityZone": "us-west-2c",
            "DbiResourceId": "db-A2GIKUV6KPOHITGGKI2NHVISZA"
        }
    ]
}
```

자세한 내용은 Amazon DocumentDB 개발자 안내서의 Amazon DocumentDB 인스턴스 설명을 참조하세요. Amazon DocumentDB

- 자세한 API 내용은 명령 참조DescribeDbInstances의 섹션을 참조하세요. AWS CLI

### `describe-db-subnet-groups`

다음 코드 예시에서는 `describe-db-subnet-groups`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon DocumentDB 서브넷 설명 목록을 검색하려면

다음 `describe-db-subnet-groups` 예제에서는 이름이 인 Amazon DocumentDB 서브넷에 대한 세부 정보를 설명합니다`default`.

```
aws docdb describe-db-subnet-groups \
    --db-subnet-group-name default
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon DocumentDB 개발자 안내서의 서브넷 그룹 설명을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeDbSubnetGroups의 섹션을 참조하세요. AWS CLI

### `describe-engine-default-cluster-parameters`

다음 코드 예시에서는 `describe-engine-default-cluster-parameters`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon DocumentDB의 기본 엔진 및 시스템 파라미터 정보를 설명하려면

다음 `describe-engine-default-cluster-parameters` 예제에서는 Amazon DocumentDB 파라미터 그룹 의 기본 엔진 및 시스템 파라미터 정보에 대한 세부 정보를 표시합니다`docdb3.6`.

```
aws docdb describe-engine-default-cluster-parameters \
    --db-parameter-group-family docdb3.6
```

출력:

```
{
    "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
            }
        ]
    }
}
```

자세한 내용은 Amazon DocumentDB 개발자 안내서DescribeEngineDefaultClusterParameters의 섹션을 참조하세요. Amazon DocumentDB

- 자세한 API 내용은 명령 참조DescribeEngineDefaultClusterParameters의 섹션을 참조하세요. AWS CLI

### `describe-event-categories`

다음 코드 예시에서는 `describe-event-categories`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 Amazon DocumentDB 이벤트 범주를 설명하려면

다음 `describe-event-categories` 예제에서는 Amazon DocumentDB 이벤트 소스 유형 의 모든 범주를 나열합니다`db-instance`.

```
aws docdb describe-event-categories \
    --source-type db-cluster
```

출력:

```
{
    "EventCategoriesMapList": [
```

```
        {
            "SourceType": "db-cluster",
            "EventCategories": [
                "failover",
                "maintenance",
                "notification",
                "failure"
            ]
        }
    ]
}
```

자세한 내용은 Amazon DocumentDB 개발자 안내서의 이벤트 범주 보기를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeEventCategories의 섹션을 참조하세요. AWS CLI

### `describe-events`

다음 코드 예시에서는 `describe-events`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon DocumentDB 이벤트를 나열하려면

다음 `describe-events` 예제에서는 지난 24시간(1,440분) 동안의 모든 Amazon DocumentDB 이벤트를 나열합니다.

```
aws docdb describe-events \
    --duration 1440
```

이 명령은 출력을 생성하지 않습니다. 출력:

```
{
    "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"
        }
```

```
    ]
}
```

자세한 내용은 Amazon DocumentDB 개발자 안내서의 Amazon DocumentDB 이벤트 보기를 참조하세요. Amazon DocumentDB

- 자세한 API 내용은 명령 참조DescribeEvents의 섹션을 참조하세요. AWS CLI

### `describe-orderable-db-instance-options`

다음 코드 예시에서는 `describe-orderable-db-instance-options`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon DocumentDB 인스턴스 옵션을 찾으려면

다음 `describe-orderable-db-instance-options` 예제에서는 리전에 대한 Amazon DocumentDB의 모든 인스턴스 옵션을 나열합니다.

```
aws docdb describe-orderable-db-instance-options \
    --engine docdb \
    --region us-east-1
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon DocumentDB 개발자 안내서의 클러스터에 Amazon DocumentDB 인스턴스 추가를 참조하세요. Amazon DocumentDB

- 자세한 API 내용은 명령 참조DescribeOrderableDbInstanceOptions의 섹션을 참조하세요. AWS CLI

### `describe-pending-maintenance-actions`

다음 코드 예시에서는 `describe-pending-maintenance-actions`을 사용하는 방법을 보여 줍니다.

AWS CLI

보류 중인 Amazon DocumentDB 유지 관리 작업을 나열하려면

다음 describe-pending-maintenance-actions 예제에서는 보류 중인 모든 Amazon DocumentDB 유지 관리 작업을 나열합니다.

```
aws docdb describe-pending-maintenance-actions
```

출력:

```
{
    "PendingMaintenanceActions": []
}
```

자세한 내용은 Amazon DocumentDB 개발자 안내서의 Amazon DocumentDB 유지 관리를 참조하세요. Amazon DocumentDB

- 자세한 API 내용은 명령 참조DescribePendingMaintenanceActions의 섹션을 참조하세요. AWS CLI

### failover-db-cluster

다음 코드 예시에서는 failover-db-cluster을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon DocumentDB 클러스터가 복제본으로 장애 조치를 수행하도록 강제 적용하려면

다음 failover-db-cluster 예제에서는 Amazon DocumentDB 클러스터 샘플 클러스터의 기본 인스턴스가 복제본으로 장애 조치합니다.

```
aws docdb failover-db-cluster \
    --db-cluster-identifier sample-cluster
```

출력:

```
{
    "DBCluster": {
        "AssociatedRoles": [],
        "DBClusterIdentifier": "sample-cluster",
        "EngineVersion": "3.6.0",
        "DBSubnetGroup": "default",
        "MasterUsername": "master-user",
        "EarliestRestorableTime": "2019-03-15T20:30:47.020Z",
```

```
        "Endpoint": "sample-cluster.cluster-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
        "AvailabilityZones": [
            "us-west-2a",
            "us-west-2c",
            "us-west-2b"
        ],
        "LatestRestorableTime": "2019-03-18T21:35:23.548Z",
        "PreferredMaintenanceWindow": "sat:04:30-sat:05:00",
        "PreferredBackupWindow": "00:00-00:30",
        "Port": 27017,
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-77186e0d",
                "Status": "active"
            }
        ],
        "StorageEncrypted": false,
        "ClusterCreateTime": "2019-03-15T20:29:58.836Z",
        "MultiAZ": true,
        "Status": "available",
        "DBClusterMembers": [
            {
                "DBClusterParameterGroupStatus": "in-sync",
                "IsClusterWriter": false,
                "DBInstanceIdentifier": "sample-cluster",
                "PromotionTier": 1
            },
            {
                "DBClusterParameterGroupStatus": "in-sync",
                "IsClusterWriter": true,
                "DBInstanceIdentifier": "sample-cluster2",
                "PromotionTier": 2
            }
        ],
        "EnabledCloudwatchLogsExports": [
            "audit"
        ],
        "DBClusterParameterGroup": "default.docdb3.6",
        "HostedZoneId": "ZNKXH85TT8WVW",
        "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster",
        "BackupRetentionPeriod": 3,
        "DbClusterResourceId": "cluster-UP4EF2PVDDFVHHDJQTYDAIGHLE",
```

```
        "ReaderEndpoint": "sample-cluster.cluster-ro-corcjozrlsfc.us-
west-2.docdb.amazonaws.com",
        "Engine": "docdb"
    }
}
```

자세한 내용은 Amazon DocumentDB 개발자 안내서의 Amazon DocumentDB 장애 조치를 참조하세요. Amazon DocumentDB

- 자세한 API 내용은 명령 참조FailoverDbCluster의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon DocumentDB 리소스의 모든 태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 Amazon DocumentDB 클러스터 의 모든 태그를 나열합니다`sample-cluster`.

```
aws docdb list-tags-for-resource \
    --resource-name arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster
```

출력:

```
{
    "TagList": [
        {
            "Key": "A",
            "Value": "ALPHA"
        },
        {
            "Key": "B",
            "Value": ""
        },
        {
            "Key": "C",
            "Value": "CHARLIE"
        }
    ]
```

```
}
```

자세한 내용은 Amazon DocumentDB 개발자 안내서의 Amazon DocumentDB 리소스에 태그 나열을 참조하세요. Amazon DocumentDB

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### `modify-db-cluster-parameter-group`

다음 코드 예시에서는 `modify-db-cluster-parameter-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon DocumentDB DB 클러스터 파라미터 그룹을 수정하려면

다음 `modify-db-cluster-parameter-group` 예제에서는 두 파라미터 `audit_logs` 및 를 활성화로 설정`custom3-6-param-grp`하여 Amazon DocumentDB 클러스터 파라미터 그룹을 수정`ttl_monitor`합니다. 변경 사항은 다음 재부팅 시 적용됩니다.

```
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
```

출력:

```
{
    "DBClusterParameterGroupName": "custom3-6-param-grp"
}
```

자세한 내용은 Amazon DocumentDB 개발자 안내서의 Amazon DocumentDB 클러스터 파라미터 그룹 수정을 참조하세요. Amazon DocumentDB

- 자세한 API 내용은 명령 참조ModifyDbClusterParameterGroup의 섹션을 참조하세요. AWS CLI

### `modify-db-cluster-snapshot-attribute`

다음 코드 예시에서는 `modify-db-cluster-snapshot-attribute`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: Amazon DocumentDB 스냅샷에 속성을 추가하려면

다음 `modify-db-cluster-snapshot-attribute` 예제에서는 Amazon DocumentDB 클러스터 스냅샷에 4개의 속성 값을 추가합니다.

```
aws docdb modify-db-cluster-snapshot-attribute \
    --db-cluster-snapshot-identifier sample-cluster-snapshot \
    --attribute-name restore \
    --values-to-add 123456789011 123456789012 123456789013
```

출력:

```
{
    "DBClusterSnapshotAttributesResult": {
        "DBClusterSnapshotAttributes": [
            {
                "AttributeName": "restore",
                "AttributeValues": [
                    "123456789011",
                    "123456789012",
                    "123456789013"
                ]
            }
        ],
        "DBClusterSnapshotIdentifier": "sample-cluster-snapshot"
    }
}
```

예제 2: Amazon DocumentDB 스냅샷에서 속성을 제거하려면

다음 `modify-db-cluster-snapshot-attribute` 예제에서는 Amazon DocumentDB 클러스터 스냅샷에서 두 개의 속성 값을 제거합니다.

```
aws docdb modify-db-cluster-snapshot-attribute \
    --db-cluster-snapshot-identifier sample-cluster-snapshot \
    --attribute-name restore \
    --values-to-remove 123456789012
```

출력:

```
{
    "DBClusterSnapshotAttributesResult": {
        "DBClusterSnapshotAttributes": [
            {
                "AttributeName": "restore",
                "AttributeValues": [
                    "123456789011",
                    "123456789013"
                ]
            }
        ],
        "DBClusterSnapshotIdentifier": "sample-cluster-snapshot"
    }
}
```

자세한 내용은 Amazon DocumentDB 개발자 안내서의 ModifyDBClusterSnapshotAttribute을 참조하세요.

- 자세한 API 내용은 명령 참조ModifyDbClusterSnapshotAttribute의 섹션을 참조하세요. AWS CLI

### `modify-db-cluster`

다음 코드 예시에서는 `modify-db-cluster`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon DocumentDB 클러스터를 수정하려면

다음 `modify-db-cluster` 예제에서는 자동 백업의 보존 기간을 7일로 설정하고 백업 및 유지 관리 모두에 대해 선호하는 기간을 변경`sample-cluster`하여 Amazon DocumentDB 클러스터를 수정합니다. 모든 변경 사항은 다음 유지 관리 기간에 적용됩니다.

```
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
```

출력:

```
{
```

```
    "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
    }
}
```

자세한 내용은 Amazon DocumentDB 개발자 안내서의 Amazon DocumentDB 클러스터 수정을 참조하세요. Amazon DocumentDB

- 자세한 API 내용은 명령 참조ModifyDbCluster의 섹션을 참조하세요. AWS CLI

### `modify-db-instance`

다음 코드 예시에서는 `modify-db-instance`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon DocumentDB 인스턴스를 수정하려면

다음 `modify-db-instance` 예제에서는 인스턴스 클래스를 로 변경`db.r4.4xlarge`하고 승격 계층을 로 변경`sample-cluster2`하여 Amazon DocumentDB 인스턴스를 수정합니다`5`. 변경 사항은 즉시 적용되지만 인스턴스 상태를 사용할 수 있는 후에만 확인할 수 있습니다.

```
aws docdb modify-db-instance \
    --db-instance-identifier sample-cluster2 \
    --apply-immediately \
    --db-instance-class db.r4.4xlarge \
    --promotion-tier 5
```

출력:

```
{
    "DBInstance": {
        "EngineVersion": "3.6.0",
        "StorageEncrypted": false,
        "DBInstanceClass": "db.r4.large",
        "PreferredMaintenanceWindow": "mon:08:39-mon:09:09",
```

```
        "AutoMinorVersionUpgrade": true,
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-77186e0d",
                "Status": "active"
            }
        ],
        "PreferredBackupWindow": "18:00-18:30",
        "EnabledCloudwatchLogsExports": [
            "audit"
        ],
        "AvailabilityZone": "us-west-2f",
        "DBInstanceIdentifier": "sample-cluster2",
        "InstanceCreateTime": "2019-03-15T20:36:06.338Z",
        "Engine": "docdb",
        "BackupRetentionPeriod": 7,
        "DBSubnetGroup": {
            "DBSubnetGroupName": "default",
            "DBSubnetGroupDescription": "default",
            "SubnetGroupStatus": "Complete",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-4e26d263",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2a"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-afc329f4",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2c"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-53ab3636",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2d"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-991cb8d0",
```

```
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2b"
                    },
                    "SubnetStatus": "Active"
                }
            ],
            "VpcId": "vpc-91280df6"
        },
        "PromotionTier": 2,
        "Endpoint": {
            "Address": "sample-cluster2.corcjozrlsfc.us-west-2.docdb.amazonaws.com",
            "HostedZoneId": "ZNKXH85TT8WVW",
            "Port": 27017
        },
        "DbiResourceId": "db-A2GIKUV6KPOHITGGKI2NHVISZA",
        "DBClusterIdentifier": "sample-cluster",
        "DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
        "PendingModifiedValues": {
            "DBInstanceClass": "db.r4.4xlarge"
        },
        "PubliclyAccessible": false,
        "DBInstanceStatus": "available"
    }
}
```

자세한 내용은 Amazon DocumentDB 개발자 안내서의 Amazon DocumentDB 인스턴스 수정을 참조하세요. Amazon DocumentDB

- 자세한 API 내용은 명령 참조ModifyDbInstance의 섹션을 참조하세요. AWS CLI

### `modify-db-subnet-group`

다음 코드 예시에서는 `modify-db-subnet-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon DocumentDB 서브넷 그룹을 수정하려면

다음 `modify-db-subnet-group` 예제에서는 지정된 서브넷과 새 설명을 `sample-subnet-group` 추가하여 서브넷 그룹을 수정합니다.

```
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"
```

출력:

```
{
    "DBSubnetGroup": {
        "DBSubnetGroupName": "sample-subnet-group",
        "SubnetGroupStatus": "Complete",
        "DBSubnetGroupArn": "arn:aws:rds:us-west-2:123456789012:subgrp:sample-
subnet-group",
        "VpcId": "vpc-91280df6",
        "DBSubnetGroupDescription": "New subnet description",
        "Subnets": [
            {
                "SubnetIdentifier": "subnet-b3806e8f",
                "SubnetStatus": "Active",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2a"
                }
            },
            {
                "SubnetIdentifier": "subnet-53ab3636",
                "SubnetStatus": "Active",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2c"
                }
            },
            {
                "SubnetIdentifier": "subnet-991cb8d0",
                "SubnetStatus": "Active",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2b"
                }
            }
        ]
    }
}
```

자세한 내용은 Amazon DocumentDB 개발자 안내서의 Amazon DocumentDB 서브넷 그룹 수정을 참조하세요. Amazon DocumentDB

- 자세한 API 내용은 명령 참조ModifyDbSubnetGroup의 섹션을 참조하세요. AWS CLI

### reboot-db-instance

다음 코드 예시에서는 reboot-db-instance을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon DocumentDB 인스턴스를 재부팅하려면

다음 reboot-db-instance 예제에서는 Amazon DocumentDB 인스턴스를 재부팅합니다sample-cluster2.

```
aws docdb reboot-db-instance \
    --db-instance-identifier sample-cluster2
```

이 명령은 출력을 생성하지 않습니다. 출력:

```
{
    "DBInstance": {
        "PreferredBackupWindow": "18:00-18:30",
        "DBInstanceIdentifier": "sample-cluster2",
        "VpcSecurityGroups": [
            {
                "Status": "active",
                "VpcSecurityGroupId": "sg-77186e0d"
            }
        ],
        "DBSubnetGroup": {
            "VpcId": "vpc-91280df6",
            "Subnets": [
                {
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2a"
                    },
                    "SubnetIdentifier": "subnet-4e26d263"
                },
                {
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2c"
                    },
                    "SubnetIdentifier": "subnet-afc329f4"
                },
```

```
                {
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2d"
                    },
                    "SubnetIdentifier": "subnet-53ab3636"
                },
                {
                    "SubnetStatus": "Active",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2b"
                    },
                    "SubnetIdentifier": "subnet-991cb8d0"
                }
            ],
            "SubnetGroupStatus": "Complete",
            "DBSubnetGroupName": "default",
            "DBSubnetGroupDescription": "default"
        },
        "PendingModifiedValues": {},
        "Endpoint": {
            "Address": "sample-cluster2.corcjozrlsfc.us-west-2.docdb.amazonaws.com",
            "HostedZoneId": "ZNKXH85TT8WVW",
            "Port": 27017
        },
        "EnabledCloudwatchLogsExports": [
            "audit"
        ],
        "StorageEncrypted": false,
        "DbiResourceId": "db-A2GIKUV6KPOHITGGKI2NHVISZA",
        "AutoMinorVersionUpgrade": true,
        "Engine": "docdb",
        "InstanceCreateTime": "2019-03-15T20:36:06.338Z",
        "EngineVersion": "3.6.0",
        "PromotionTier": 5,
        "BackupRetentionPeriod": 7,
        "DBClusterIdentifier": "sample-cluster",
        "PreferredMaintenanceWindow": "mon:08:39-mon:09:09",
        "PubliclyAccessible": false,
        "DBInstanceClass": "db.r4.4xlarge",
        "AvailabilityZone": "us-west-2d",
        "DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:sample-cluster2",
        "DBInstanceStatus": "rebooting"
    }
```

```
}
```

자세한 내용은 Amazon DocumentDB 개발자 안내서의 Amazon DocumentDB 재부팅ILnstance을 참조하세요. Amazon DocumentDB

- 자세한 API 내용은 명령 참조RebootDbInstance의 섹션을 참조하세요. AWS CLI

### `remove-tags-from-resource`

다음 코드 예시에서는 `remove-tags-from-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon DocumentDB 리소스에서 태그를 제거하려면

다음 `remove-tags-from-resource` 예제에서는 Amazon DocumentDB 클러스터 B에서 이름이인 키가 있는 태그를 제거합니다`sample-cluster`.

```
aws docdb remove-tags-from-resource \
    --resource-name arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster \
    --tag-keys B
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon DocumentDB 개발자 안내서의 Amazon D에서 태그 제거 ocumentDBResource를 참조하세요. Amazon DocumentDB

- 자세한 API 내용은 명령 참조RemoveTagsFromResource의 섹션을 참조하세요. AWS CLI

### `reset-db-cluster-parameter-group`

다음 코드 예시에서는 `reset-db-cluster-parameter-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon DocumentDB 파라미터 그룹에서 지정된 파라미터 값을 기본값으로 재설정하려면

다음 `reset-db-cluster-parameter-group` 예제에서는 `ttl_monitor` Amazon DocumentDB 파라미터 그룹의 파라미터를 기본값`custom3-6-param-grp`으로 재설정합니다.

```
aws docdb reset-db-cluster-parameter-group \
```

```
    --db-cluster-parameter-group-name custom3-6-param-grp \
    --parameters ParameterName=ttl_monitor,ApplyMethod=immediate
```

출력:

```
{
    "DBClusterParameterGroupName": "custom3-6-param-grp"
}
```

자세한 내용은 Amazon DocumentDB 개발자 안내서의 제목을 참조하세요.

Amazon DocumentDB 파라미터 그룹에서 지정된 파라미터 값 또는 모든 파라미터 값을 기본값으로 재설정하려면

다음 `reset-db-cluster-parameter-group` 예제에서는 Amazon DocumentDB 파라미터 그룹의 모든 파라미터를 기본값`custom3-6-param-grp`으로 재설정합니다.

```
aws docdb reset-db-cluster-parameter-group \
    --db-cluster-parameter-group-name custom3-6-param-grp \
    --reset-all-parameters
```

출력:

```
{
    "DBClusterParameterGroupName": "custom3-6-param-grp"
}
```

자세한 내용은 Amazon DocumentDB 개발자 안내서의 Amazon DocumentDB 클러스터 파라미터 그룹 재설정을 참조하세요. Amazon DocumentDB

- 자세한 API 내용은 명령 참조ResetDbClusterParameterGroup의 섹션을 참조하세요. AWS CLI

### `restore-db-cluster-from-snapshot`

다음 코드 예시에서는 `restore-db-cluster-from-snapshot`을 사용하는 방법을 보여 줍니다.

AWS CLI

자동 또는 수동 스냅샷에서 Amazon DocumentDB 클러스터를 복원하려면

다음 `restore-db-cluster-from-snapshot` 예제에서는 스냅샷 `sample-cluster-2019-03-16-00-01-restored`에서 라는 새 Amazon DocumentDB 클러스터를 생성합니다`rds:sample-cluster-2019-03-16-00-01`.

```
aws docdb restore-db-cluster-from-snapshot \
    --db-cluster-identifier sample-cluster-2019-03-16-00-01-restored \
    --engine docdb \
    --snapshot-identifier rds:sample-cluster-2019-03-16-00-01
```

출력:

```
{
    "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-
corcjozrlsfc.us-west-2.docdb.amazonaws.com",
        "Endpoint": "sample-cluster-2019-03-16-00-01-restored.cluster-
corcjozrlsfc.us-west-2.docdb.amazonaws.com",
        "Port": 27017,
        "PreferredBackupWindow": "00:00-00:30",
        "DBSubnetGroup": "default",
        "DBClusterIdentifier": "sample-cluster-2019-03-16-00-01-restored",
        "PreferredMaintenanceWindow": "sat:04:30-sat:05:00",
        "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-
cluster-2019-03-16-00-01-restored",
        "DBClusterParameterGroup": "default.docdb3.6",
        "DbClusterResourceId": "cluster-XOO46Q3RH4LWSYNH3NMZKXPISU",
        "MasterUsername": "master-user",
        "EngineVersion": "3.6.0",
        "BackupRetentionPeriod": 3,
        "AssociatedRoles": [],
        "Status": "creating",
        "VpcSecurityGroups": [
```

```
            {
                "Status": "active",
                "VpcSecurityGroupId": "sg-77186e0d"
            }
        ]
    }
}
```

자세한 내용은 Amazon DocumentDB 개발자 안내서의 클러스터 스냅샷에서 복원을 참조하세요.

- 자세한 API 내용은 명령 참조RestoreDbClusterFromSnapshot의 섹션을 참조하세요. AWS CLI

### `restore-db-cluster-to-point-in-time`

다음 코드 예시에서는 `restore-db-cluster-to-point-in-time`을 사용하는 방법을 보여 줍니다.

AWS CLI

수동 스냅샷에서 Amazon DocumentDB 클러스터를 point-in-time로 복원하려면

다음 `restore-db-cluster-to-point-in-time` 예제에서는 `sample-cluster-snapshot`를 사용하여 최신 복원 가능 시간을 사용하여 새 Amazon DocumentDB 클러스터 `sample-cluster-pit`를 생성합니다.

```
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
```

출력:

```
{
    "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-NLCABBXOSE2QPQ4GOLZIFWEPLM",
        "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster-
pit"
    }
}
```

자세한 내용은 Amazon DocumentDB 개발자 안내서의 특정 시점으로 스냅샷 복원을 참조하세요.

- 자세한 API 내용은 명령 참조RestoreDbClusterToPointInTime의 섹션을 참조하세요. AWS CLI

### `start-db-cluster`

다음 코드 예시에서는 `start-db-cluster`을 사용하는 방법을 보여 줍니다.

AWS CLI

중지된 Amazon DocumentDB 클러스터를 시작하려면

다음 start-db-cluster 예제에서는 지정된 Amazon DocumentDB 클러스터를 시작합니다.

```
aws docdb start-db-cluster \
    --db-cluster-identifier sample-cluster
```

출력:

```
{
    "DBCluster": {
        "ClusterCreateTime": "2019-03-19T18:45:01.857Z",
        "HostedZoneId": "ZNKXH85TT8WVW",
        "Engine": "docdb",
        "DBClusterMembers": [],
        "MultiAZ": false,
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1c",
            "us-east-1f"
        ],
        "StorageEncrypted": false,
        "ReaderEndpoint": "sample-cluster-2019-03-16-00-01-restored.cluster-ro-
corcjozrlsfc.us-east-1.docdb.amazonaws.com",
        "Endpoint": "sample-cluster-2019-03-16-00-01-restored.cluster-
corcjozrlsfc.us-east-1.docdb.amazonaws.com",
        "Port": 27017,
        "PreferredBackupWindow": "00:00-00:30",
        "DBSubnetGroup": "default",
        "DBClusterIdentifier": "sample-cluster-2019-03-16-00-01-restored",
        "PreferredMaintenanceWindow": "sat:04:30-sat:05:00",
        "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:sample-
cluster-2019-03-16-00-01-restored",
        "DBClusterParameterGroup": "default.docdb3.6",
        "DbClusterResourceId": "cluster-XOO46Q3RH4LWSYNH3NMZKXPISU",
        "MasterUsername": "master-user",
        "EngineVersion": "3.6.0",
        "BackupRetentionPeriod": 3,
        "AssociatedRoles": [],
        "Status": "creating",
        "VpcSecurityGroups": [
            {
                "Status": "active",
                "VpcSecurityGroupId": "sg-77186e0d"
            }
```

```
        ]
    }
}
```

자세한 내용은 Amazon DocumentDB 개발자 안내서의 Amazon DocumentDB 클러스터 중지 및 시작을 참조하세요. Amazon DocumentDB

- 자세한 API 내용은 명령 참조StartDbCluster의 섹션을 참조하세요. AWS CLI

### stop-db-cluster

다음 코드 예시에서는 stop-db-cluster을 사용하는 방법을 보여 줍니다.

AWS CLI

실행 중인 Amazon DocumentDB 클러스터를 중지하려면

다음 stop-db-cluster 예제에서는 지정된 Amazon DocumentDB 클러스터를 중지합니다.

```
aws docdb stop-db-cluster \
    --db-cluster-identifier sample-cluster
```

출력:

```
{
    "DBCluster": {
        "ClusterCreateTime": "2019-03-19T18:45:01.857Z",
        "HostedZoneId": "ZNKXH85TT8WVW",
        "Engine": "docdb",
        "DBClusterMembers": [],
        "MultiAZ": false,
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1c",
            "us-east-1f"
        ],
        "StorageEncrypted": false,
        "ReaderEndpoint": "sample-cluster-2019-03-16-00-01-restored.cluster-ro-
corcjozrlsfc.us-east-1.docdb.amazonaws.com",
        "Endpoint": "sample-cluster-2019-03-16-00-01-restored.cluster-
corcjozrlsfc.us-east-1.docdb.amazonaws.com",
        "Port": 27017,
```

```
        "PreferredBackupWindow": "00:00-00:30",
        "DBSubnetGroup": "default",
        "DBClusterIdentifier": "sample-cluster-2019-03-16-00-01-restored",
        "PreferredMaintenanceWindow": "sat:04:30-sat:05:00",
        "DBClusterArn": "arn:aws:rds:us-east-1:123456789012:cluster:sample-
cluster-2019-03-16-00-01-restored",
        "DBClusterParameterGroup": "default.docdb3.6",
        "DbClusterResourceId": "cluster-XOO46Q3RH4LWSYNH3NMZKXPISU",
        "MasterUsername": "master-user",
        "EngineVersion": "3.6.0",
        "BackupRetentionPeriod": 3,
        "AssociatedRoles": [],
        "Status": "creating",
        "VpcSecurityGroups": [
            {
                "Status": "active",
                "VpcSecurityGroupId": "sg-77186e0d"
            }
        ]
    }
}
```

자세한 내용은 Amazon DocumentDB 개발자 안내서의 Amazon DocumentDB 클러스터 중지 및 시작을 참조하세요. Amazon DocumentDB

- 자세한 API 내용은 명령 참조StopDbCluster의 섹션을 참조하세요. AWS CLI

## 를 사용한 DynamoDB 예제 AWS CLI

다음 코드 예제에서는 DynamoDB와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### `batch-get-item`

다음 코드 예시에서는 batch-get-item을 사용하는 방법을 보여 줍니다.

AWS CLI

테이블에서 여러 항목을 검색하는 방법

다음 batch-get-items 예시에서는 GetItem 요청 3개의 배치를 사용하여 MusicCollection 테이블에서 여러 항목을 읽고 작업에 사용된 읽기 용량 단위 수를 요청합니다. 이 명령은 AlbumTitle 속성만 반환합니다.

```
aws dynamodb batch-get-item \
    --request-items file://request-items.json \
    --return-consumed-capacity TOTAL
```

request-items.json의 콘텐츠:

```
{
    "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"
    }
}
```

출력:

```
{
```

```
    "Responses": {
        "MusicCollection": [
            {
                "AlbumTitle": {
                    "S": "Somewhat Famous"
                }
            },
            {
                "AlbumTitle": {
                    "S": "Blue Sky Blues"
                }
            },
            {
                "AlbumTitle": {
                    "S": "Louder Than Ever"
                }
            }
        ]
    },
    "UnprocessedKeys": {},
    "ConsumedCapacity": [
        {
            "TableName": "MusicCollection",
            "CapacityUnits": 1.5
        }
    ]
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 배치 작업을 참조하세요.

- 자세한 API 내용은 명령 참조BatchGetItem의 섹션을 참조하세요. AWS CLI

### batch-write-item

다음 코드 예시에서는 batch-write-item을 사용하는 방법을 보여 줍니다.

AWS CLI

테이블에 여러 항목을 추가하는 방법

다음 batch-write-item 예시에서는 PutItem 요청 3개의 배치를 사용하여 MusicCollection 테이블에 새 항목 3개를 추가합니다. 또한 작업에 사용된 쓰기 용량 단위 수와 작업에서 수정된 모든 항목 모음에 대한 정보도 요청합니다.

```
aws dynamodb batch-write-item \
    --request-items file://request-items.json \
    --return-consumed-capacity INDEXES \
    --return-item-collection-metrics SIZE
```

request-items.json의 콘텐츠:

```
{
    "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"}
                }
            }
        }
    ]
}
```

출력:

```
{
```

```
    "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
                }
            }
        }
    ]
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 배치 작업을 참조하세요.

- 자세한 API 내용은 명령 참조BatchWriteItem의 섹션을 참조하세요. AWS CLI

### create-backup

다음 코드 예시에서는 create-backup을 사용하는 방법을 보여 줍니다.

AWS CLI

기존 DynamoDB 테이블에 대한 백업을 생성하려면

다음 create-backup 예제에서는 MusicCollection 테이블의 백업을 생성합니다.

```
aws dynamodb create-backup \
    --table-name MusicCollection \
    --backup-name MusicCollectionBackup
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB용 온디맨드 백업 및 복원을 참조하세요. DynamoDB

- 자세한 API 내용은 명령 참조CreateBackup의 섹션을 참조하세요. AWS CLI

### create-global-table

다음 코드 예시에서는 create-global-table을 사용하는 방법을 보여 줍니다.

AWS CLI

글로벌 테이블을 생성하려면

다음 `create-global-table` 예제에서는 지정된 별도의 AWS 리전에 있는 두 개의 동일한 테이블에서 전역 테이블을 생성합니다.

```
aws dynamodb create-global-table \
    --global-table-name MusicCollection \
    --replication-group RegionName=us-east-2 RegionName=us-east-1 \
    --region us-east-2
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB 글로벌 테이블을 참조하세요. DynamoDB

- 자세한 API 내용은 명령 참조CreateGlobalTable의 섹션을 참조하세요. AWS CLI

## create-table

다음 코드 예시에서는 `create-table`을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: 태그가 포함된 테이블을 생성하는 방법

다음 create-table 예시에서는 지정된 속성과 키 스키마를 사용하여 이름이 MusicCollection인 테이블을 생성합니다. 이 테이블은 프로비저닝된 처리량을 사용하며 기본 AWS 소유 를 사용하여 저장 시 암호화됩니다CMK. 이 명령은 또한 키가 Owner이고 값이 blueTeam인 태그를 테이블에 적용합니다.

```
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
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 테이블 기본 작업을 참조하세요.

예 2: 온디맨드 모드에서 테이블을 생성하는 방법

다음 예시에서는 프로비저닝된 처리량 모드가 아닌 온디맨드 모드를 사용하여 이름이 MusicCollection인 테이블을 생성합니다. 이는 예상치 못한 워크로드가 있는 테이블에 유용합니다.

```
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
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 테이블 기본 작업을 참조하세요.

예제 3: 고객 관리형 테이블을 생성하고 암호화하려면 CMK

다음 예제에서는 라는 테이블을 생성하고 고객 관리형 MusicCollection 를 사용하여 암호화합니다CMK.

```
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
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 테이블 기본 작업을 참조하세요.

예 4: 로컬 보조 인덱스가 있는 테이블을 생성하는 방법

다음 MusicCollection 예시에서는 지정된 속성과 키 스키마를 사용하여 이름이 AlbumTitleIndex인 로컬 보조 인덱스가 있는 이라는 테이블을 생성합니다.

```
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\"]
                }
            }
        ]"
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 테이블 기본 작업을 참조하세요.

예 5: 글로벌 보조 인덱스가 있는 테이블을 생성하는 방법

다음 예시에서는 이름이 GameTitleIndex인 글로벌 보조 인덱스가 있는 GameScores라는 테이블을 생성합니다. 기본 테이블에는 의 파티션 키UserId와 의 정렬 키가 GameTitle있어 특정 게임에 대한 개별 사용자의 최고 점수를 효율적으로 찾을 수 있는 반면, GSI에는 의 파티션 키GameTitle와 의 정렬 키가 TopScore있어 특정 게임에 대한 전체 최고 점수를 빠르게 찾을 수 있습니다.

```
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
            }
        }
    ]"
```

출력:

```
{
    "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"
            }
        ]
    }
```

```
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 테이블 기본 작업을 참조하세요.

예 6: 글로벌 보조 인덱스가 있는 테이블 여러 개를 한 번에 생성하는 방법

다음 예시에서는 두 개의 글로벌 보조 인덱스가 있는 GameScores라는 테이블을 생성합니다. GSI 스키마는 명령줄이 아닌 파일을 통해 전달됩니다.

```
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
```

gsi.json의 콘텐츠:

```
[
    {
        "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
        }
    }
]
```

출력:

```
{
    "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"
            }
        ]
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 테이블 기본 작업을 참조하세요.

예 7: Streams가 활성화된 테이블을 생성하는 방법

다음 예시에서는 DynamoDB Streams가 활성화된 GameScores라는 테이블을 생성합니다. 각 항목의 새 이미지와 이전 이미지가 모두 스트림에 작성됩니다.

```
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
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 테이블 기본 작업을 참조하세요.

예 8: Keys-Only Stream이 활성화된 테이블을 생성하는 방법

다음 예시에서는 DynamoDB Streams가 활성화된 GameScores라는 테이블을 생성합니다. 수정된 항목의 키 속성만 스트림에 작성됩니다.

```
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
```

출력:

```
{
    "TableDescription": {
        "AttributeDefinitions": [
            {
                "AttributeName": "GameTitle",
                "AttributeType": "S"
            },
            {
                "AttributeName": "UserId",
                "AttributeType": "S"
            }
        ],
```

```
        "TableName": "GameScores",
        "KeySchema": [
            {
                "AttributeName": "UserId",
                "KeyType": "HASH"
            },
            {
                "AttributeName": "GameTitle",
                "KeyType": "RANGE"
            }
        ],
        "TableStatus": "CREATING",
        "CreationDateTime": "2023-05-25T18:45:34.140000+00:00",
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 10,
            "WriteCapacityUnits": 5
        },
        "TableSizeBytes": 0,
        "ItemCount": 0,
        "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/GameScores",
        "TableId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "StreamSpecification": {
            "StreamEnabled": true,
            "StreamViewType": "KEYS_ONLY"
        },
        "LatestStreamLabel": "2023-05-25T18:45:34.140",
        "LatestStreamArn": "arn:aws:dynamodb:us-west-2:123456789012:table/
GameScores/stream/2023-05-25T18:45:34.140",
        "DeletionProtectionEnabled": false
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB Streams에 대한 변경 데이터 캡처를 참조하세요.

예 9: Standard-Infrequent Access 클래스를 사용하는 테이블을 생성하는 방법

다음 예시에서는 이름이 GameScores인 테이블을 생성하고 Standard-Infrequent Access(DynamoDB Standard-IA) 테이블 클래스를 할당합니다. 이 테이블 클래스는 가장 비용이 많이 드는 스토리지에 최적화되어 있습니다.

```
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
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 테이블 클래스를 참조하세요.

예 10: 삭제 방지가 활성화된 테이블을 생성하는 방법

다음 예시에서는 이름이 GameScores인 테이블을 생성하고 삭제 방지를 활성화합니다.

```
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
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 삭제 보호 기능 사용을 참조하세요.

- 자세한 API 내용은 명령 참조CreateTable의 섹션을 참조하세요. AWS CLI

### `delete-backup`

다음 코드 예시에서는 `delete-backup`을 사용하는 방법을 보여 줍니다.

AWS CLI

기존 DynamoDB 백업을 삭제하려면

다음 `delete-backup` 예제에서는 지정된 기존 백업을 삭제합니다.

```
aws dynamodb delete-backup \
    --backup-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection/
backup/01576616366715-b4e58d3a
```

출력:

```
{
    "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": {}
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB용 온디맨드 백업 및 복원을 참조하세요. DynamoDB

- 자세한 API 내용은 명령 참조DeleteBackup의 섹션을 참조하세요. AWS CLI

### delete-item

다음 코드 예시에서는 delete-item을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: 항목을 삭제하는 방법

다음 delete-item 예시에서는 MusicCollection 테이블에서 항목을 삭제하고 삭제된 항목에 대한 세부 정보와 요청에 사용된 용량을 요청합니다.

```
aws dynamodb delete-item \
    --table-name MusicCollection \
    --key file://key.json \
    --return-values ALL_OLD \
    --return-consumed-capacity TOTAL \
    --return-item-collection-metrics SIZE
```

key.json의 콘텐츠:

```
{
    "Artist": {"S": "No One You Know"},
    "SongTitle": {"S": "Scared of My Shadow"}
}
```

출력:

```
{
    "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
        ]
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 항목 쓰기를 참조하세요.

예 2: 조건부로 항목을 삭제하는 방법

다음 예시에서는 ProductCategory가 Sporting Goods 또는 Gardening Supplies이고 가격이 500에서 600 사이일 때만 ProductCatalog 테이블에서 항목을 삭제합니다. 삭제된 항목에 대한 세부 정보가 반환됩니다.

```
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
```

names.json의 콘텐츠:

```
{
    "#P": "Price"
}
```

values.json의 콘텐츠:

```
{
    ":cat1": {"S": "Sporting Goods"},
    ":cat2": {"S": "Gardening Supplies"},
```

```
    ":lo": {"N": "500"},
    ":hi": {"N": "600"}
}
```

출력:

```
{
    "Attributes": {
        "Id": {
            "N": "456"
        },
        "Price": {
            "N": "550"
        },
        "ProductCategory": {
            "S": "Sporting Goods"
        }
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 항목 쓰기를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteItem의 섹션을 참조하세요. AWS CLI

### `delete-table`

다음 코드 예시에서는 `delete-table`을 사용하는 방법을 보여 줍니다.

AWS CLI

테이블을 삭제하는 방법

다음 `delete-table` 예시에서는 `MusicCollection` 테이블을 삭제합니다.

```
aws dynamodb delete-table \
    --table-name MusicCollection
```

출력:

```
{
    "TableDescription": {
        "TableStatus": "DELETING",
```

```
        "TableSizeBytes": 0,
        "ItemCount": 0,
        "TableName": "MusicCollection",
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
            "WriteCapacityUnits": 5,
            "ReadCapacityUnits": 5
        }
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 테이블 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteTable의 섹션을 참조하세요. AWS CLI

### `describe-backup`

다음 코드 예시에서는 `describe-backup`을 사용하는 방법을 보여 줍니다.

AWS CLI

테이블의 기존 백업에 대한 정보를 가져오려면

다음 `describe-backup` 예제에서는 지정된 기존 백업에 대한 정보를 표시합니다.

```
aws dynamodb describe-backup \
    --backup-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection/
backup/01576616366715-b4e58d3a
```

출력:

```
{
    "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": {}
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB용 온디맨드 백업 및 복원을 참조하세요. DynamoDB

- 자세한 API 내용은 명령 참조DescribeBackup의 섹션을 참조하세요. AWS CLI

### `describe-continuous-backups`

다음 코드 예시에서는 `describe-continuous-backups`을 사용하는 방법을 보여 줍니다.

AWS CLI

DynamoDB 테이블의 연속 백업에 대한 정보를 가져오려면

다음 `describe-continuous-backups` 예제에서는 `MusicCollection` 테이블의 연속 백업 설정에 대한 세부 정보를 표시합니다.

```
aws dynamodb describe-continuous-backups \
    --table-name MusicCollection
```

출력:

```
{
    "ContinuousBackupsDescription": {
        "ContinuousBackupsStatus": "ENABLED",
        "PointInTimeRecoveryDescription": {
            "PointInTimeRecoveryStatus": "DISABLED"
        }
    }
}
```

자세한 내용은 Amazon Point-in-Time DynamoDB 개발자 안내서의 DynamoDB에 대한 복구를 참조하세요. DynamoDB

- 자세한 API 내용은 명령 참조DescribeContinuousBackups의 섹션을 참조하세요. AWS CLI

### describe-contributor-insights

다음 코드 예시에서는 describe-contributor-insights을 사용하는 방법을 보여 줍니다.

AWS CLI

DynamoDB 테이블에 대한 Contributor Insights 설정을 보려면

다음 describe-contributor-insights 예제에서는 MusicCollection 테이블 및 AlbumTitle-index 글로벌 보조 인덱스에 대한 Contributor Insights 설정을 표시합니다.

```
aws dynamodb describe-contributor-insights \
    --table-name MusicCollection \
    --index-name AlbumTitle-index
```

출력:

```
{
    "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
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB용 CloudWatch Contributor Insights를 사용하여 데이터 액세스 분석을 참조하세요. DynamoDB

- 자세한 API 내용은 명령 참조DescribeContributorInsights의 섹션을 참조하세요. AWS CLI

### `describe-endpoints`

다음 코드 예시에서는 `describe-endpoints`을 사용하는 방법을 보여 줍니다.

AWS CLI

리전 엔드포인트 정보를 보려면

다음 `describe-endpoints` 예제에서는 현재 AWS 리전의 엔드포인트에 대한 세부 정보를 표시합니다.

```
aws dynamodb describe-endpoints
```

출력:

```
{
    "Endpoints": [
        {
            "Address": "dynamodb.us-west-2.amazonaws.com",
            "CachePeriodInMinutes": 1440
        }
    ]
}
```

자세한 내용은 AWS 일반 참조 의 Amazon DynamoDB 엔드포인트 및 할당량을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeEndpoints의 섹션을 참조하세요. AWS CLI

### describe-global-table-settings

다음 코드 예시에서는 describe-global-table-settings을 사용하는 방법을 보여 줍니다.

AWS CLI

DynamoDB 글로벌 테이블 설정에 대한 정보를 가져오려면

다음 describe-global-table-settings 예제에서는 MusicCollection 전역 테이블에 대한 설정을 표시합니다.

```
aws dynamodb describe-global-table-settings \
    --global-table-name MusicCollection
```

출력:

```
{
    "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
            }
        }
```

```
    ]
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB 글로벌 테이블을 참조하세요. DynamoDB

- 자세한 API 내용은 명령 참조DescribeGlobalTableSettings의 섹션을 참조하세요. AWS CLI

### `describe-global-table`

다음 코드 예시에서는 `describe-global-table`을 사용하는 방법을 보여 줍니다.

AWS CLI

DynamoDB 글로벌 테이블에 대한 정보를 표시하려면

다음 `describe-global-table` 예제에서는 `MusicCollection` 전역 테이블에 대한 세부 정보를 보여줍니다.

```
aws dynamodb describe-global-table \
    --global-table-name MusicCollection
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB 글로벌 테이블을 참조하세요. DynamoDB

- 자세한 API 내용은 명령 참조DescribeGlobalTable의 섹션을 참조하세요. AWS CLI

### describe-limits

다음 코드 예시에서는 describe-limits을 사용하는 방법을 보여 줍니다.

AWS CLI

프로비저닝된 용량 제한을 보려면

다음 describe-limits 예제에서는 현재 AWS 리전의 계정에 대해 프로비저닝된 용량 제한을 보여줍니다.

```
aws dynamodb describe-limits
```

출력:

```
{
    "AccountMaxReadCapacityUnits": 80000,
    "AccountMaxWriteCapacityUnits": 80000,
    "TableMaxReadCapacityUnits": 40000,
    "TableMaxWriteCapacityUnits": 40000
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB의 제한을 참조하세요. DynamoDB

- 자세한 API 내용은 명령 참조DescribeLimits의 섹션을 참조하세요. AWS CLI

### describe-table-replica-auto-scaling

다음 코드 예시에서는 describe-table-replica-auto-scaling을 사용하는 방법을 보여 줍니다.

AWS CLI

글로벌 테이블의 복제본 간에 자동 크기 조정 설정을 보려면

다음 describe-table-replica-auto-scaling 예제에서는 MusicCollection 전역 테이블의 복제본에 대한 자동 조정 설정을 표시합니다.

```
aws dynamodb describe-table-replica-auto-scaling \
    --table-name MusicCollection
```

출력:

```
{
    "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"
            }
```

```
        ]
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB 글로벌 테이블을 참조하세요.
DynamoDB

- 자세한 API 내용은 명령 참조DescribeTableReplicaAutoScaling의 섹션을 참조하세요. AWS CLI

### describe-table

다음 코드 예시에서는 describe-table을 사용하는 방법을 보여 줍니다.

AWS CLI

테이블을 설명하는 방법

다음 describe-table 예시에서는 MusicCollection 테이블을 설명합니다.

```
aws dynamodb describe-table \
    --table-name MusicCollection
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 테이블 설명을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeTable의 섹션을 참조하세요. AWS CLI

### describe-time-to-live

다음 코드 예시에서는 describe-time-to-live을 사용하는 방법을 보여 줍니다.

AWS CLI

테이블의 Time to Live 설정을 보려면

다음 describe-time-to-live 예제에서는 MusicCollection 테이블의 Time to Live 설정을 표시합니다.

```
aws dynamodb describe-time-to-live \
    --table-name MusicCollection
```

출력:

```
{
    "TimeToLiveDescription": {
        "TimeToLiveStatus": "ENABLED",
        "AttributeName": "ttl"
    }
```

```
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 Time to Live를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeTimeToLive의 섹션을 참조하세요. AWS CLI

### `get-item`

다음 코드 예시에서는 `get-item`을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: 테이블의 항목을 읽는 방법

다음 `get-item` 예시에서는 `MusicCollection` 테이블에서 항목을 검색합니다. 테이블에는 기본 키(`Artist` 및 `SongTitle`)가 hash-and-range 있으므로 이러한 속성을 모두 지정해야 합니다. 또한 이 명령은 작업에 사용된 읽기 용량에 대한 정보를 요청합니다.

```
aws dynamodb get-item \
    --table-name MusicCollection \
    --key file://key.json \
    --return-consumed-capacity TOTAL
```

`key.json`의 콘텐츠:

```
{
    "Artist": {"S": "Acme Band"},
    "SongTitle": {"S": "Happy Day"}
}
```

출력:

```
{
    "Item": {
        "AlbumTitle": {
            "S": "Songs About Life"
        },
        "SongTitle": {
            "S": "Happy Day"
        },
        "Artist": {
```

```
            "S": "Acme Band"
        }
    },
    "ConsumedCapacity": {
        "TableName": "MusicCollection",
        "CapacityUnits": 0.5
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 항목 읽기를 참조하세요.

예 2: 일관된 읽기를 사용하여 항목을 읽는 방법

다음 예시에서는 강력히 일관되 읽기를 사용하여 `MusicCollection` 테이블의 항목을 읽습니다.

```
aws dynamodb get-item \
    --table-name MusicCollection \
    --key file://key.json \
    --consistent-read \
    --return-consumed-capacity TOTAL
```

`key.json`의 콘텐츠:

```
{
    "Artist": {"S": "Acme Band"},
    "SongTitle": {"S": "Happy Day"}
}
```

출력:

```
{
    "Item": {
        "AlbumTitle": {
            "S": "Songs About Life"
        },
        "SongTitle": {
            "S": "Happy Day"
        },
        "Artist": {
            "S": "Acme Band"
        }
    },
```

```
    "ConsumedCapacity": {
        "TableName": "MusicCollection",
        "CapacityUnits": 1.0
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 항목 읽기를 참조하세요.

예 3: 항목의 특정 속성을 검색하는 방법

다음 예시에서는 프로젝션 표현식을 사용하여 원하는 항목의 세 가지 속성만 검색합니다.

```
aws dynamodb get-item \
    --table-name ProductCatalog \
    --key '{"Id": {"N": "102"}}' \
    --projection-expression "#T, #C, #P" \
    --expression-attribute-names file://names.json
```

names.json의 콘텐츠:

```
{
    "#T": "Title",
    "#C": "ProductCategory",
    "#P": "Price"
}
```

출력:

```
{
    "Item": {
        "Price": {
            "N": "20"
        },
        "Title": {
            "S": "Book 102 Title"
        },
        "ProductCategory": {
            "S": "Book"
        }
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 항목 읽기를 참조하세요.

- 자세한 API 내용은 명령 참조GetItem의 섹션을 참조하세요. AWS CLI

### `list-backups`

다음 코드 예시에서는 `list-backups`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 기존 DynamoDB 백업을 모두 나열하려면

다음 `list-backups` 예제에서는 기존 백업을 모두 나열합니다.

```
aws dynamodb list-backups
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB용 온디맨드 백업 및 복원을 참조하세요. DynamoDB

예제 2: 특정 시간 범위에서 사용자가 생성한 백업을 나열하려면

다음 예제에서는 생성 날짜가 2020년 1월 1일부터 2020년 3월 1일 사이인 사용자( DynamoDB에서 자동으로 생성한 백업 제외)가 생성한 MusicCollection 테이블의 백업만 나열합니다.

```
aws dynamodb list-backups \
    --table-name MusicCollection \
    --time-range-lower-bound 1577836800 \
    --time-range-upper-bound 1583020800 \
    --backup-type USER
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB용 온디맨드 백업 및 복원을 참조하세요. DynamoDB

예제 3: 페이지 크기 제한

다음 예제에서는 기존 백업 목록을 모두 반환하지만, 각 호출에서 하나의 항목만 검색하여 전체 목록을 가져오는 데 필요한 경우 여러 호출을 수행합니다. 페이지 크기 제한은 많은 리소스에서 list 명령을 실행할 때 유용합니다. 리소스가 많을 때 기본 페이지 크기인 1,000을 사용하면 '시간 초과' 오류가 발생할 수 있습니다.

```
aws dynamodb list-backups \
    --page-size 1
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB용 온디맨드 백업 및 복원을 참조하세요. DynamoDB

예제 4: 반환된 항목 수를 제한하려면

다음 예제에서는 반환되는 항목 수를 1로 제한합니다. 응답에는 결과의 다음 페이지를 검색하는 데 사용되는 NextToken 값이 포함됩니다.

```
aws dynamodb list-backups \
    --max-items 1
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB용 온디맨드 백업 및 복원을 참조하세요. DynamoDB

예제 5: 결과의 다음 페이지를 검색하려면

다음 명령은 이전의 list-backups 명령 호출에서 얻은 NextToken 값을 사용하여 다른 결과 페이지를 검색합니다. 이 경우 응답에는 NextToken 값이 포함되어 있지 않으므로 결과의 끝에 도달했음을 알 수 있습니다.

```
aws dynamodb list-backups \
    --starting-
token abCDeFGhiJKlmnOPqrSTuvwxYZ1aBCdEFghijK7LM51nOpqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9
```

출력

```
{
    "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
        }
    ]
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB용 온디맨드 백업 및 복원을 참조하세요. DynamoDB

- 자세한 API 내용은 명령 참조ListBackups의 섹션을 참조하세요. AWS CLI

### `list-contributor-insights`

다음 코드 예시에서는 `list-contributor-insights`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: Contributor Insights 요약 목록을 보려면

다음 `list-contributor-insights` 예제에서는 Contributor Insights 요약 목록을 보여줍니다.

```
aws dynamodb list-contributor-insights
```

출력:

```
{
    "ContributorInsightsSummaries": [
        {
            "TableName": "MusicCollection",
            "IndexName": "AlbumTitle-index",
            "ContributorInsightsStatus": "ENABLED"
        },
        {
            "TableName": "ProductCatalog",
            "ContributorInsightsStatus": "ENABLED"
        },
        {
            "TableName": "Forum",
            "ContributorInsightsStatus": "ENABLED"
        },
        {
            "TableName": "Reply",
            "ContributorInsightsStatus": "ENABLED"
        },
        {
            "TableName": "Thread",
            "ContributorInsightsStatus": "ENABLED"
        }
    ]
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB용 CloudWatch Contributor Insights를 사용하여 데이터 액세스 분석을 참조하세요. DynamoDB

예제 2: 반환된 항목 수를 제한하려면

다음 예제에서는 반환되는 항목 수를 4개로 제한합니다. 응답에는 결과의 다음 페이지를 검색하는 데 사용되는 NextToken 값이 포함됩니다.

```
aws dynamodb list-contributor-insights \
    --max-results 4
```

출력:

```
{
    "ContributorInsightsSummaries": [
        {
```

```
            "TableName": "MusicCollection",
            "IndexName": "AlbumTitle-index",
            "ContributorInsightsStatus": "ENABLED"
        },
        {
            "TableName": "ProductCatalog",
            "ContributorInsightsStatus": "ENABLED"
        },
        {
            "TableName": "Forum",
            "ContributorInsightsStatus": "ENABLED"
        }
    ],
    "NextToken":
 "abCDeFGhiJKlmnOPqrSTuvwxYZ1aBCdEFghijK7LM51nOpqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9"
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB용 CloudWatch Contributor Insights를 사용하여 데이터 액세스 분석을 참조하세요. DynamoDB

예제 3: 결과의 다음 페이지를 검색하려면

다음 명령은 이전의 `list-contributor-insights` 명령 호출에서 얻은 `NextToken` 값을 사용하여 다른 결과 페이지를 검색합니다. 이 경우 응답에는 `NextToken` 값이 포함되어 있지 않으므로 결과의 끝에 도달했음을 알 수 있습니다.

```
aws dynamodb list-contributor-insights \
    --max-results 4 \
    --next-
token abCDeFGhiJKlmnOPqrSTuvwxYZ1aBCdEFghijK7LM51nOpqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9
```

출력:

```
{
    "ContributorInsightsSummaries": [
        {
            "TableName": "Reply",
            "ContributorInsightsStatus": "ENABLED"
        },
        {
            "TableName": "Thread",
            "ContributorInsightsStatus": "ENABLED"
```

```
        }
    ]
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB용 CloudWatch Contributor Insights를 사용하여 데이터 액세스 분석을 참조하세요. DynamoDB

- 자세한 API 내용은 명령 참조ListContributorInsights의 섹션을 참조하세요. AWS CLI

### `list-global-tables`

다음 코드 예시에서는 `list-global-tables`을 사용하는 방법을 보여 줍니다.

AWS CLI

기존 DynamoDB 글로벌 테이블을 나열하려면

다음 `list-global-tables` 예제에서는 기존 글로벌 테이블을 모두 나열합니다.

```
aws dynamodb list-global-tables
```

출력:

```
{
    "GlobalTables": [
        {
            "GlobalTableName": "MusicCollection",
            "ReplicationGroup": [
                {
                    "RegionName": "us-east-2"
                },
                {
                    "RegionName": "us-east-1"
                }
            ]
        }
    ]
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB 글로벌 테이블을 참조하세요. DynamoDB

- 자세한 API 내용은 명령 참조ListGlobalTables의 섹션을 참조하세요. AWS CLI

### `list-tables`

다음 코드 예시에서는 `list-tables`을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: 테이블을 나열하는 방법

다음 `list-tables` 예제에서는 현재 AWS 계정 및 리전과 연결된 모든 테이블을 나열합니다.

```
aws dynamodb list-tables
```

출력:

```
{
    "TableNames": [
        "Forum",
        "ProductCatalog",
        "Reply",
        "Thread"
    ]
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 테이블 이름 나열을 참조하세요.

예 2: 페이지 크기를 제한하는 방법

다음 예시에서는 모든 기존 테이블의 목록을 반환하지만 각 호출에서 항목을 하나만 검색하고, 필요한 경우 전체 목록을 가져오기 위해 여러 번 호출합니다. 페이지 크기 제한은 많은 리소스에서 list 명령을 실행할 때 유용합니다. 리소스가 많을 때 기본 페이지 크기인 1,000을 사용하면 '시간 초과' 오류가 발생할 수 있습니다.

```
aws dynamodb list-tables \
    --page-size 1
```

출력:

```
{
    "TableNames": [
        "Forum",
        "ProductCatalog",
        "Reply",
```

```
        "Thread"
    ]
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 테이블 이름 나열을 참조하세요.

예 3: 반환되는 항목 수를 제한하는 방법

다음 예시에서는 반환되는 항목 수를 2개로 제한합니다. 응답에는 결과의 다음 페이지를 검색하는 데 사용되는 NextToken 값이 포함됩니다.

```
aws dynamodb list-tables \
    --max-items 2
```

출력:

```
{
    "TableNames": [
        "Forum",
        "ProductCatalog"
    ],
    "NextToken":
 "abCDeFGhiJKlmnOPqrSTuvwxYZ1aBCdEFghijK7LM51nOpqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9"
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 테이블 이름 나열을 참조하세요.

예 4: 결과의 다음 페이지를 검색하는 방법

다음 명령은 이전의 list-tables 명령 호출에서 얻은 NextToken 값을 사용하여 다른 결과 페이지를 검색합니다. 이 경우 응답에는 NextToken 값이 포함되어 있지 않으므로 결과의 끝에 도달했음을 알 수 있습니다.

```
aws dynamodb list-tables \
    --starting-
token abCDeFGhiJKlmnOPqrSTuvwxYZ1aBCdEFghijK7LM51nOpqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9
```

출력:

```
{
    "TableNames": [
```

```
        "Reply",
        "Thread"
    ]
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 테이블 이름 나열을 참조하세요.

- 자세한 API 내용은 명령 참조ListTables의 섹션을 참조하세요. AWS CLI

### `list-tags-of-resource`

다음 코드 예시에서는 `list-tags-of-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: DynamoDB 리소스의 태그를 나열하려면

다음 `list-tags-of-resource` 예제에서는 `MusicCollection` 테이블에 대한 태그를 표시합니다.

```
aws dynamodb list-tags-of-resource \
    --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection
```

출력:

```
{
    "Tags": [
        {
            "Key": "Owner",
            "Value": "blueTeam"
        },
        {
            "Key": "Environment",
            "Value": "Production"
        }
    ]
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB에 대한 태그 지정을 참조하세요. DynamoDB

예제 2: 반환되는 태그 수 제한

다음 예제에서는 반환되는 태그 수를 1로 제한합니다. 응답에는 결과의 다음 페이지를 검색하는 데 사용되는 NextToken 값이 포함됩니다.

```
aws dynamodb list-tags-of-resource \
    --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection \
    --max-items 1
```

출력:

```
{
    "Tags": [
        {
            "Key": "Owner",
            "Value": "blueTeam"
        }
    ],
    "NextToken":
 "abCDeFGhiJKlmnOPqrSTuvwxYZ1aBCdEFghijK7LM51nOpqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9"
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB에 대한 태그 지정을 참조하세요.

예제 3: 결과의 다음 페이지를 검색하려면

다음 명령은 이전의 list-tags-of-resource 명령 호출에서 얻은 NextToken 값을 사용하여 다른 결과 페이지를 검색합니다. 이 경우 응답에는 NextToken 값이 포함되어 있지 않으므로 결과의 끝에 도달했음을 알 수 있습니다.

```
aws dynamodb list-tags-of-resource \
    --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection \
    --starting-
token abCDeFGhiJKlmnOPqrSTuvwxYZ1aBCdEFghijK7LM51nOpqRSTuv3WxY3ZabC5dEFGhI2Jk3LmnoPQ6RST9
```

출력:

```
{
    "Tags": [
        {
            "Key": "Environment",
            "Value": "Production"
```

```
        }
    ]
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB에 대한 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조ListTagsOfResource의 섹션을 참조하세요. AWS CLI

### put-item

다음 코드 예시에서는 put-item을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: 테이블에 항목을 추가하는 방법

다음 put-item 예제에서는 MusicCollection 테이블에 새 항목을 추가합니다.

```
aws dynamodb put-item \
    --table-name MusicCollection \
    --item file://item.json \
    --return-consumed-capacity TOTAL \
    --return-item-collection-metrics SIZE
```

item.json의 콘텐츠:

```
{
    "Artist": {"S": "No One You Know"},
    "SongTitle": {"S": "Call Me Today"},
    "AlbumTitle": {"S": "Greatest Hits"}
}
```

출력:

```
{
    "ConsumedCapacity": {
        "TableName": "MusicCollection",
        "CapacityUnits": 1.0
    },
    "ItemCollectionMetrics": {
        "ItemCollectionKey": {
```

```
            "Artist": {
                "S": "No One You Know"
            }
        },
        "SizeEstimateRangeGB": [
            0.0,
            1.0
        ]
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 항목 쓰기를 참조하세요.

예 2: 테이블의 항목을 조건부로 덮어쓰는 방법

다음 put-item 예시에서는 기존 항목에 값이 Greatest Hits인 AlbumTitle 속성이 있는 경우에만 MusicCollection 테이블의 기존 항목을 덮어씁니다. 이 명령은 항목의 이전 값을 반환합니다.

```
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
```

item.json의 콘텐츠:

```
{
    "Artist": {"S": "No One You Know"},
    "SongTitle": {"S": "Call Me Today"},
    "AlbumTitle": {"S": "Somewhat Famous"}
}
```

names.json의 콘텐츠:

```
{
    "#A": "AlbumTitle"
}
```

values.json의 콘텐츠:

```
{
    ":A": {"S": "Greatest Hits"}
}
```

출력:

```
{
    "Attributes": {
        "AlbumTitle": {
            "S": "Greatest Hits"
        },
        "Artist": {
            "S": "No One You Know"
        },
        "SongTitle": {
            "S": "Call Me Today"
        }
    }
}
```

키가 이미 있는 경우 다음과 같은 출력이 표시됩니다.

```
A client error (ConditionalCheckFailedException) occurred when calling the PutItem
 operation: The conditional request failed.
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 항목 쓰기를 참조하세요.

- 자세한 API 내용은 명령 참조PutItem의 섹션을 참조하세요. AWS CLI

### `query`

다음 코드 예시에서는 `query`을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: 테이블을 쿼리하는 방법

다음 `query` 예시에서는 `MusicCollection` 테이블의 항목을 쿼리합니다. 테이블에는 기본 키(`Artist` 및 `SongTitle`)가 hash-and-range 있지만 이 쿼리는 해시 키 값만 지정합니다. 'No One You Know'라는 아티스트의 노래 제목이 반환됩니다.

```
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
```

expression-attributes.json의 콘텐츠:

```
{
    ":v1": {"S": "No One You Know"}
}
```

출력:

```
{
    "Items": [
        {
            "SongTitle": {
                "S": "Call Me Today"
            },
            "SongTitle": {
                "S": "Scared of My Shadow"
            }
        }
    ],
    "Count": 2,
    "ScannedCount": 2,
    "ConsumedCapacity": {
        "TableName": "MusicCollection",
        "CapacityUnits": 0.5
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB에서 쿼리 작업을 참조하세요.

예 2: 강력히 일관된 읽기를 사용하여 테이블을 쿼리하고 인덱스를 내림차순으로 탐색하는 방법

다음 예시에서는 첫 번째 예와 동일한 쿼리를 수행하지만 결과를 역순으로 반환하고 강력히 일관된 읽기를 사용합니다.

```
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
```

expression-attributes.json의 콘텐츠:

```
{
    ":v1": {"S": "No One You Know"}
}
```

출력:

```
{
    "Items": [
        {
            "SongTitle": {
                "S": "Scared of My Shadow"
            }
        },
        {
            "SongTitle": {
                "S": "Call Me Today"
            }
        }
    ],
    "Count": 2,
    "ScannedCount": 2,
    "ConsumedCapacity": {
        "TableName": "MusicCollection",
        "CapacityUnits": 1.0
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB에서 쿼리 작업을 참조하세요.

예 3: 특정 결과를 필터링하는 방법

다음 예시에서는 MusicCollection을 쿼리하되 AlbumTitle 속성에 특정 값이 있는 결과를 제외합니다. 항목을 읽은 후에 필터가 적용되므로 ScannedCount 또는 ConsumedCapacity에는 영향을 주지 않는다는 점에 유의하세요.

```
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
```

values.json의 콘텐츠:

```
{
    ":v1": {"S": "No One You Know"},
    ":v2": {"S": "Blue Sky Blues"},
    ":v3": {"S": "Greatest Hits"}
}
```

names.json의 콘텐츠:

```
{
    "#n1": "Artist",
    "#n2": "AlbumTitle"
}
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB에서 쿼리 작업을 참조하세요.

예 4: 항목 수만 검색하는 방법

다음 예시에서는 쿼리와 일치하는 항목 수를 검색하지만 항목 자체는 검색하지 않습니다.

```
aws dynamodb query \
    --table-name MusicCollection \
    --select COUNT \
    --key-condition-expression "Artist = :v1" \
    --expression-attribute-values file://expression-attributes.json
```

expression-attributes.json의 콘텐츠:

```
{
    ":v1": {"S": "No One You Know"}
}
```

출력:

```
{
    "Count": 2,
    "ScannedCount": 2,
    "ConsumedCapacity": null
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB에서 쿼리 작업을 참조하세요.

예 5: 인덱스를 쿼리하는 방법

다음 예시에서는 로컬 보조 인덱스 AlbumTitleIndex를 쿼리합니다. 쿼리는 로컬 보조 인덱스로 프로젝션된 기본 테이블의 모든 속성을 반환합니다. 로컬 보조 인덱스 또는 글로벌 보조 인덱스를 쿼리할 때는 table-name 파라미터를 사용하여 기본 테이블의 이름도 제공해야 한다는 점에 유의하세요.

```
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
```

expression-attributes.json의 콘텐츠:

```
{
    ":v1": {"S": "No One You Know"}
}
```

출력:

```
{
    "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
            }
        }
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB에서 쿼리 작업을 참조하세요.

- 자세한 API 내용은 AWS CLI 명령 참조의 쿼리를 참조하세요.

### `restore-table-from-backup`

다음 코드 예시에서는 `restore-table-from-backup`을 사용하는 방법을 보여 줍니다.

AWS CLI

기존 백업에서 DynamoDB 테이블을 복원하려면

다음 `restore-table-from-backup` 예제는 기존 백업에서 지정된 테이블을 복원합니다.

```
aws dynamodb restore-table-from-backup \
    --target-table-name MusicCollection \
    --backup-arnarn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection/
backup/01576616366715-b4e58d3a
```

출력:

```
{
    "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
        }
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB용 온디맨드 백업 및 복원을 참조하세요. DynamoDB

- 자세한 API 내용은 명령 참조RestoreTableFromBackup의 섹션을 참조하세요. AWS CLI

### restore-table-to-point-in-time

다음 코드 예시에서는 restore-table-to-point-in-time을 사용하는 방법을 보여 줍니다.

AWS CLI

DynamoDB 테이블을 특정 시점으로 복원하려면

다음 restore-table-to-point-in-time 예제에서는 MusicCollection 테이블을 지정된 시점으로 복원합니다.

```
aws dynamodb restore-table-to-point-in-time \
    --source-table-name MusicCollection \
    --target-table-name MusicCollectionRestore \
    --restore-date-time 1576622404.0
```

출력:

```
{
    "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
        }
    }
}
```

자세한 내용은 Amazon Point-in-Time DynamoDB 개발자 안내서의 DynamoDB용 복구를 참조하세요. DynamoDB

- 자세한 API 내용은 명령 참조RestoreTableToPointInTime의 섹션을 참조하세요. AWS CLI

### scan

다음 코드 예시에서는 scan을 사용하는 방법을 보여 줍니다.

AWS CLI

테이블을 스캔하는 방법

다음 scan 예시에서는 MusicCollection 테이블 전체를 스캔한 다음 'No One You Know' 아티스트의 곡으로 결과 범위를 좁힙니다. 각 항목에 대해 앨범 제목과 노래 제목만 반환됩니다.

```
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
```

expression-attribute-names.json의 콘텐츠:

```
{
    "#ST": "SongTitle",
    "#AT":"AlbumTitle"
}
```

expression-attribute-values.json의 콘텐츠:

```
{
    ":a": {"S": "No One You Know"}
}
```

출력:

```
{
    "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
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB에서 스캔 작업을 참조하세요.

- 자세한 API 내용은 명령 참조의 스캔을 참조하세요. AWS CLI

### tag-resource

다음 코드 예시에서는 tag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

DynamoDB 리소스에 태그를 추가하려면

다음 tag-resource 예제에서는 태그 키/값 페어를 MusicCollection 테이블에 추가합니다.

```
aws dynamodb tag-resource \
    --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection \
    --tags Key=Owner,Value=blueTeam
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB에 대한 태그 지정을 참조하세요. DynamoDB

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### transact-get-items

다음 코드 예시에서는 transact-get-items을 사용하는 방법을 보여 줍니다.

AWS CLI

하나 이상의 테이블에서 여러 항목을 원자적으로 검색하려면

다음 transact-get-items 예제에서는 여러 항목을 원자적으로 검색합니다.

```
aws dynamodb transact-get-items \
    --transact-items file://transact-items.json \
    --return-consumed-capacity TOTAL
```

transact-items.json의 콘텐츠:

```
[
    {
        "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"
        }
    }
]
```

출력:

```
{
    "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"
                }
            }
        }
    ]
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB 트랜잭션을 사용한 복잡한 워크플로 관리를 참조하세요. DynamoDB

- 자세한 API 내용은 명령 참조TransactGetItems의 섹션을 참조하세요. AWS CLI

### `transact-write-items`

다음 코드 예시에서는 `transact-write-items`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 항목을 하나 이상의 테이블에 원자적으로 작성하려면

다음 `transact-write-items` 예제에서는 한 항목을 업데이트하고 다른 항목을 삭제합니다. 작업이 실패하거나 항목에 `Rating` 속성이 포함된 경우 작업이 실패합니다.

```
aws dynamodb transact-write-items \
    --transact-items file://transact-items.json \
    --return-consumed-capacity TOTAL \
    --return-item-collection-metrics SIZE
```

`transact-items.json` 파일 내용:

```
[
```

```
    {
        "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)"
        }
    }
]
```

출력:

```
{
    "ConsumedCapacity": [
        {
            "TableName": "MusicCollection",
            "CapacityUnits": 10.0,
            "WriteCapacityUnits": 10.0
        }
    ],
    "ItemCollectionMetrics": {
        "MusicCollection": [
            {
                "ItemCollectionKey": {
                    "Artist": {
                        "S": "No One You Know"
                    }
```

```
                },
                "SizeEstimateRangeGB": [
                    0.0,
                    1.0
                ]
            },
            {
                "ItemCollectionKey": {
                    "Artist": {
                        "S": "Acme Band"
                    }
                },
                "SizeEstimateRangeGB": [
                    0.0,
                    1.0
                ]
            }
        ]
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB 트랜잭션을 사용한 복잡한 워크플로 관리를 참조하세요. DynamoDB

예제 2: 클라이언트 요청 토큰을 사용하여 항목을 원자적으로 작성하려면

다음 명령은 클라이언트 요청 토큰을 사용하여 `transact-write-items` idempotent로 호출합니다. 즉, 여러 호출이 하나의 단일 호출과 동일한 효과를 갖습니다.

```
aws dynamodb transact-write-items \
    --transact-items file://transact-items.json \
    --client-request-token abc123
```

`transact-items.json` 파일 내용:

```
[
    {
        "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)"
        }
    }
]
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB 트랜잭션을 사용한 복잡한 워크플로 관리를 참조하세요. DynamoDB

- 자세한 API 내용은 명령 참조TransactWriteItems의 섹션을 참조하세요. AWS CLI

### `untag-resource`

다음 코드 예시에서는 `untag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

DynamoDB 리소스에서 태그를 제거하려면

다음 `untag-resource` 예제에서는 `MusicCollection` 테이블`Owner`에서 키가 있는 태그를 제거합니다.

```
aws dynamodb untag-resource \
    --resource-arn arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection \
    --tag-keys Owner
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB에 대한 태그 지정을 참조하세요. DynamoDB

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-continuous-backups

다음 코드 예시에서는 update-continuous-backups을 사용하는 방법을 보여 줍니다.

AWS CLI

DynamoDB 테이블에 대한 연속 백업 설정을 업데이트하려면

다음 update-continuous-backups 예제에서는 MusicCollection 테이블에 대한 복구를 활성화합니다 point-in-time.

```
aws dynamodb update-continuous-backups \
    --table-name MusicCollection \
    --point-in-time-recovery-specification PointInTimeRecoveryEnabled=true
```

출력:

```
{
    "ContinuousBackupsDescription": {
        "ContinuousBackupsStatus": "ENABLED",
        "PointInTimeRecoveryDescription": {
            "PointInTimeRecoveryStatus": "ENABLED",
            "EarliestRestorableDateTime": 1576622404.0,
            "LatestRestorableDateTime": 1576622404.0
        }
    }
}
```

자세한 내용은 Amazon Point-in-Time DynamoDB 개발자 안내서의 DynamoDB용 복구를 참조하세요. DynamoDB

- 자세한 API 내용은 명령 참조UpdateContinuousBackups의 섹션을 참조하세요. AWS CLI

### update-contributor-insights

다음 코드 예시에서는 update-contributor-insights을 사용하는 방법을 보여 줍니다.

AWS CLI

테이블에서 Contributor Insights를 활성화하려면

다음 `update-contributor-insights` 예제에서는 `MusicCollection` 테이블과 `AlbumTitle-index` 글로벌 보조 인덱스에서 Contributor Insights를 활성화합니다.

```
aws dynamodb update-contributor-insights \
    --table-name MusicCollection \
    --index-name AlbumTitle-index \
    --contributor-insights-action ENABLE
```

출력:

```
{
    "TableName": "MusicCollection",
    "IndexName": "AlbumTitle-index",
    "ContributorInsightsStatus": "ENABLING"
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB용 CloudWatch Contributor Insights를 사용하여 데이터 액세스 분석을 참조하세요.

- 자세한 API 내용은 AWS CLI 명령 참조의 UpdateContributorInsights 섹션을 참조하세요.

### update-global-table-settings

다음 코드 예시에서는 `update-global-table-settings`을 사용하는 방법을 보여 줍니다.

AWS CLI

DynamoDB 글로벌 테이블에서 프로비저닝된 쓰기 용량 설정을 업데이트하려면

다음 `update-global-table-settings` 예제에서는 `MusicCollection` 전역 테이블의 프로비저닝된 쓰기 용량을 15로 설정합니다.

```
aws dynamodb update-global-table-settings \
    --global-table-name MusicCollection \
    --global-table-provisioned-write-capacity-units 15
```

출력:

```
{
    "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
            }
        }
    ]
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB 글로벌 테이블을 참조하세요.
DynamoDB

- 자세한 API 내용은 명령 참조UpdateGlobalTableSettings의 섹션을 참조하세요. AWS CLI

### update-global-table

다음 코드 예시에서는 update-global-table을 사용하는 방법을 보여 줍니다.

AWS CLI

DynamoDB 글로벌 테이블을 업데이트하려면

다음 update-global-table 예제에서는 지정된 리전의 복제본을 MusicCollection 글로벌 테이블에 추가합니다.

```
aws dynamodb update-global-table \
    --global-table-name MusicCollection \
    --replica-updates Create={RegionName=eu-west-1}
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB 글로벌 테이블을 참조하세요.
DynamoDB

- 자세한 API 내용은 명령 참조UpdateGlobalTable의 섹션을 참조하세요. AWS CLI

### update-item

다음 코드 예시에서는 update-item을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: 테이블의 항목을 업데이트하는 방법

다음 update-item 예제에서는 MusicCollection 테이블의 항목을 업데이트합니다. 새 속성(Year)을 추가하고 AlbumTitle 속성을 수정합니다. 업데이트 후에 표시되는 항목 속성이 모두 응답에 반환됩니다.

```
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
```

key.json의 콘텐츠:

```
{
    "Artist": {"S": "Acme Band"},
    "SongTitle": {"S": "Happy Day"}
}
```

expression-attribute-names.json의 콘텐츠:

```
{
    "#Y":"Year", "#AT":"AlbumTitle"
}
```

expression-attribute-values.json의 콘텐츠:

```
{
    ":y":{"N": "2015"},
    ":t":{"S": "Louder Than Ever"}
}
```

출력:

```
{
    "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
        ]
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 항목 쓰기를 참조하세요.

예 2: 항목을 조건부로 업데이트하는 방법

다음 예시에서는 기존 항목에 Year 속성이 없는 경우에만 MusicCollection 테이블의 항목을 업데이트합니다.

```
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)"
```

key.json의 콘텐츠:

```
{
    "Artist": {"S": "Acme Band"},
    "SongTitle": {"S": "Happy Day"}
}
```

expression-attribute-names.json의 콘텐츠:

```
{
    "#Y":"Year",
    "#AT":"AlbumTitle"
}
```

expression-attribute-values.json의 콘텐츠:

```
{
    ":y":{"N": "2015"},
    ":t":{"S": "Louder Than Ever"}
}
```

항목에 이미 Year 속성이 있는 경우 DynamoDB는 다음 출력을 반환합니다.

```
An error occurred (ConditionalCheckFailedException) when calling the UpdateItem
 operation: The conditional request failed
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 항목 쓰기를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateItem의 섹션을 참조하세요. AWS CLI

### update-table-replica-auto-scaling

다음 코드 예시에서는 update-table-replica-auto-scaling을 사용하는 방법을 보여 줍니다.

AWS CLI

글로벌 테이블의 복제본 간에 Auto Scaling 설정을 업데이트하려면

다음 update-table-replica-auto-scaling 예제에서는 지정된 전역 테이블의 복제본 간에 쓰기 용량 자동 조정 설정을 업데이트합니다.

```
aws dynamodb update-table-replica-auto-scaling \
    --table-name MusicCollection \
    --provisioned-write-capacity-auto-scaling-update file://auto-scaling-policy.json
```

auto-scaling-policy.json의 콘텐츠:

```
{
    "MinimumUnits": 10,
    "MaximumUnits": 100,
    "AutoScalingDisabled": false,
    "ScalingPolicyUpdate": {
        "PolicyName": "DynamoDBWriteCapacityUtilization:table/MusicCollection",
        "TargetTrackingScalingPolicyConfiguration": {
            "TargetValue": 80
        }
    }
}
```

출력:

```
{
    "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"
            }
        ]
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB 글로벌 테이블을 참조하세요. DynamoDB

- 자세한 API 내용은 명령 참조UpdateTableReplicaAutoScaling의 섹션을 참조하세요. AWS CLI

### update-table

다음 코드 예시에서는 update-table을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: 테이블의 결제 모드를 수정하는 방법

다음 update-table 예시에서는 MusicCollection 테이블에 프로비저닝된 읽기 및 쓰기 용량을 늘립니다.

```
aws dynamodb update-table \
    --table-name MusicCollection \
    --billing-mode PROVISIONED \
```

```
    --provisioned-throughput ReadCapacityUnits=15,WriteCapacityUnits=10
```

출력:

```
{
    "TableDescription": {
        "AttributeDefinitions": [
            {
                "AttributeName": "AlbumTitle",
                "AttributeType": "S"
            },
            {
                "AttributeName": "Artist",
                "AttributeType": "S"
            },
            {
                "AttributeName": "SongTitle",
                "AttributeType": "S"
            }
        ],
        "TableName": "MusicCollection",
        "KeySchema": [
            {
                "AttributeName": "Artist",
                "KeyType": "HASH"
            },
            {
                "AttributeName": "SongTitle",
                "KeyType": "RANGE"
            }
        ],
        "TableStatus": "UPDATING",
        "CreationDateTime": "2020-05-26T15:59:49.473000-07:00",
        "ProvisionedThroughput": {
            "LastIncreaseDateTime": "2020-07-28T13:18:18.921000-07:00",
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 15,
            "WriteCapacityUnits": 10
        },
        "TableSizeBytes": 182,
        "ItemCount": 2,
        "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/MusicCollection",
        "TableId": "abcd0123-01ab-23cd-0123-abcdef123456",
```

```
        "BillingModeSummary": {
            "BillingMode": "PROVISIONED",
            "LastUpdateToPayPerRequestDateTime": "2020-07-28T13:14:48.366000-07:00"
        }
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 Updating a Table을 참조하세요.

예 2: 글로벌 보조 인덱스를 생성하는 방법

다음 예시에서는 MusicCollection 테이블에 글로벌 보조 인덱스를 추가합니다.

```
aws dynamodb update-table \
    --table-name MusicCollection \
    --attribute-definitions AttributeName=AlbumTitle,AttributeType=S \
    --global-secondary-index-updates file://gsi-updates.json
```

gsi-updates.json의 콘텐츠:

```
[
    {
        "Create": {
            "IndexName": "AlbumTitle-index",
            "KeySchema": [
                {
                    "AttributeName": "AlbumTitle",
                    "KeyType": "HASH"
                }
            ],
            "ProvisionedThroughput": {
                "ReadCapacityUnits": 10,
                "WriteCapacityUnits": 10
            },
            "Projection": {
                "ProjectionType": "ALL"
            }
        }
    }
]
```

출력:

```
{
    "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"
            }
        ]
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 Updating a Table을 참조하세요.

예 3: 테이블에서 DynamoDB Streams를 활성화하는 방법

다음 명령은 `MusicCollection` 테이블에서 DynamoDB Streams를 활성화합니다.

```
aws dynamodb update-table \
    --table-name MusicCollection \
    --stream-specification StreamEnabled=true,StreamViewType=NEW_IMAGE
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 Updating a Table을 참조하세요.

예 4: 서버 측 암호화를 활성화하는 방법

다음 예시에서는 MusicCollection 테이블에서 서버 측 암호화를 활성화합니다.

```
aws dynamodb update-table \
    --table-name MusicCollection \
    --sse-specification Enabled=true,SSEType=KMS
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 Updating a Table을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateTable의 섹션을 참조하세요. AWS CLI

### update-time-to-live

다음 코드 예시에서는 update-time-to-live을 사용하는 방법을 보여 줍니다.

AWS CLI

테이블에서 Time to Live 설정을 업데이트하려면

다음 update-time-to-live 예제에서는 지정된 테이블에서 Time to Live를 활성화합니다.

```
aws dynamodb update-time-to-live \
    --table-name MusicCollection \
    --time-to-live-specification Enabled=true,AttributeName=ttl
```

출력:

```
{
    "TimeToLiveSpecification": {
        "Enabled": true,
        "AttributeName": "ttl"
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 Time to Live를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateTimeToLive의 섹션을 참조하세요. AWS CLI

## 를 사용한 DynamoDB Streams 예제 AWS CLI

다음 코드 예제에서는 DynamoDB Streams와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `describe-stream`

다음 코드 예시에서는 `describe-stream`을 사용하는 방법을 보여 줍니다.

AWS CLI

DynamoDB 스트림에 대한 정보를 가져오려면

다음 `describe-stream` 명령은 특정 DynamoDB 스트림에 대한 정보를 표시합니다.

```
aws dynamodbstreams describe-stream \
    --stream-arn arn:aws:dynamodb:us-west-1:123456789012:table/Music/
stream/2019-10-22T18:02:01.576
```

출력:

```
{
    "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": "400000000000642977831",
                    "EndingSequenceNumber": "400000000000642977831"
                }
            },
            {
                "ShardId": "shardId-00000001571780995058-40810d86",
                "SequenceNumberRange": {
                    "StartingSequenceNumber": "757400000000005655171150"
                },
                "ParentShardId": "shardId-00000001571767321804-697ce3d2"
            }
        ]
    }
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB 스트림을 사용한 테이블 활동 캡처를 참조하세요. DynamoDB

- 자세한 API 내용은 명령 참조DescribeStream의 섹션을 참조하세요. AWS CLI

### get-records

다음 코드 예시에서는 get-records을 사용하는 방법을 보여 줍니다.

AWS CLI

Dynamodb 스트림에서 레코드를 가져오려면

다음 get-records 명령은 지정된 Amazon DynamoDB 샤드 반복기를 사용하여 레코드를 검색합니다.

```
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+B4EbsByi52InVmiONmy6xVW4IRcIIbs1zO7MNI
+CjNPlqQjnyRSAnfOwWmKhL1/KNParWSfz2odf780oOObIDIWRRMkt7+Hyzh9SD
+hFxFAWR5C7QIlOXPc8mRBfNIazfrVCjJK8/jsjCzsqNyXKzJbhh+GXCoxYN
+Kpmg4nyj1EAsYhbGL35muvHFoHjcyuynbsczbWaXNfThDwRAyvoTmc8XhHKtAWUbJiaVd8ZPtQwDsThCrmDRPIdmTRG
+w/lEGS05ha1qNP+Vl4+tuhz2TRnhnJo/pny9GI/yGpce97mWvSPr5KPwy+Dtcm5BHayBs
+PVYHITaTliInFlT
+LCwvaz1QH3MY3b8A05Z800wjpktm60iQqtMeDwN4NX6FrcxR34JoFKGsgR8XkHVJzz2xr1xqSJ12ycpNTyHnndusw==
```

출력:

```
{
    "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|AAAAAAAAAAEhEI04jkFLW
+LKOwivjT8d/IHEh3iExV2xK00aTxEzVy1C1C7Kbb5+ZOW6bT9VQ2n1/
mrs7+PRiaOZCHJu7JHJVW7zlsqOi/ges3fw8GYEymyL+piEk35cx67rQqwKKyq
+Q6w9JyjreIOj4F2lWLV26lBwRTrIYC4IB7C3BZZK4715QwYdDxNdVHiSBRZX8UqoS6WOt0F87xZLNB9F/
NhYBLXi/wcGvAcBcC0TNIOH+N0NqwtoB/
FGCkNrf8YZ0xRoNN6RgGuVWHF3pxOhxEJeFZoSoJTIKeG9YcYxzi5Ci/
mhdtm7tBXnbw5c6xmsGsBqTirNjlDyJLcWl8Cl0UOLX63Ufo/5QliztcjEbKsQe28x8LM8o7VH1Is0fF/
```

```
ITt8awSA4igyJS0P87GN8Qri8kj8iaE35805jBHWF2wvwT6Iy2xGrR2r2HzYps9dwGOarVdEITaJfWzNoL4HajMhmREZ
+V04i1YIeHMXJfcwetNRuIbdQXfJht2NQZa4PVV6iknY6d19MrdbSTMKoqAuvp6g3Q2jH4t7GKCLWgodcPAn8g5+43Da
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB 스트림을 사용한 테이블 활동 캡처를 참조하세요. DynamoDB

- 자세한 API 내용은 명령 참조GetRecords의 섹션을 참조하세요. AWS CLI

### get-shard-iterator

다음 코드 예시에서는 get-shard-iterator을 사용하는 방법을 보여 줍니다.

AWS CLI

샤드 반복자를 가져오려면

다음 get-shard-iterator 명령은 지정된 샤드에 대한 샤드 반복기를 검색합니다.

```
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
```

출력:

```
{
    "ShardIterator": "arn:aws:dynamodb:us-west-1:123456789012:table/Music/
stream/2019-10-22T18:02:01.576|1|
AAAAAAAAAAGgM3YZ89vLZZxjmoQeo33r9M4x3+zmmTLsiL86MfrF4+B4EbsByi52InVmiONmy6xVW4IRcIIbs1zO7MNI
+CjNPlqQjnyRSAnfOwWmKhL1/KNParWSfz2odf780oOObIDIWRRMkt7+Hyzh9SD
+hFxFAWR5C7QIlOXPc8mRBfNIazfrVCjJK8/jsjCzsqNyXKzJbhh+GXCoxYN
+Kpmg4nyj1EAsYhbGL35muvHFoHjcyuynbsczbWaXNfThDwRAyvoTmc8XhHKtAWUbJiaVd8ZPtQwDsThCrmDRPIdmTRG
+w/lEGS05ha1qNP+Vl4+tuhz2TRnhnJo/pny9GI/yGpce97mWvSPr5KPwy+Dtcm5BHayBs
+PVYHITaTliInFlT
+LCwvaz1QH3MY3b8A05Z800wjpktm60iQqtMeDwN4NX6FrcxR34JoFKGsgR8XkHVJzz2xr1xqSJ12ycpNTyHnndusw==
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB 스트림을 사용한 테이블 활동 캡처를 참조하세요. DynamoDB

- 자세한 API 내용은 명령 참조GetShardIterator의 섹션을 참조하세요. AWS CLI

**`list-streams`**

다음 코드 예시에서는 `list-streams`을 사용하는 방법을 보여 줍니다.

AWS CLI

DynamoDB 스트림을 나열하려면

다음 `list-streams` 명령은 기본 AWS 리전 내의 기존 Amazon DynamoDB 스트림을 모두 나열합니다.

```
aws dynamodbstreams list-streams
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon DynamoDB 개발자 안내서의 DynamoDB 스트림을 사용한 테이블 활동 캡처를 참조하세요. DynamoDB

- 자세한 API 내용은 명령 참조ListStreams의 섹션을 참조하세요. AWS CLI

## 를 사용한 Amazon EC2 예제 AWS CLI

다음 코드 예제에서는 Amazon 와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다EC2.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### accept-address-transfer

다음 코드 예시에서는 accept-address-transfer을 사용하는 방법을 보여 줍니다.

AWS CLI

계정으로 전송된 탄력적 IP 주소를 수락하려면

다음 accept-address-transfer 예제에서는 지정된 탄력적 IP 주소를 계정으로 전송하는 것을 허용합니다.

```
aws ec2 accept-address-transfer \
    --address 100.21.184.216
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon VPC 사용 설명서의 탄력적 IP 주소 전송을 참조하세요.

- 자세한 API 내용은 명령 참조AcceptAddressTransfer의 섹션을 참조하세요. AWS CLI

### accept-reserved-instances-exchange-quote

다음 코드 예시에서는 accept-reserved-instances-exchange-quote을 사용하는 방법을 보여 줍니다.

AWS CLI

컨버터블 예약 인스턴스 교환을 수행하려면

이 예제에서는 지정된 컨버터블 예약 인스턴스를 교환합니다.

명령:

```
aws ec2 accept-reserved-instances-exchange-quote --reserved-
instance-ids 7b8750c3-397e-4da4-bbcb-a45ebexample --target-
configurations OfferingId=b747b472-423c-48f3-8cee-679bcexample
```

출력:

```
{
  "ExchangeId": "riex-e68ed3c1-8bc8-4c17-af77-811afexample"
}
```

- 자세한 API 내용은 명령 참조AcceptReservedInstancesExchangeQuote의 섹션을 참조하세요. AWS CLI

### accept-transit-gateway-peering-attachment

다음 코드 예시에서는 accept-transit-gateway-peering-attachment을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 피어링 연결을 수락하려면

다음 accept-transit-gateway-peering-attachment 예제에서는 지정된 전송 게이트웨이 피어링 연결을 허용합니다. --region 파라미터는 수락자 전송 게이트웨이가 위치한 리전을 지정합니다.

```
aws ec2 accept-transit-gateway-peering-attachment \
    --transit-gateway-attachment-id tgw-attach-4455667788aabbccd \
    --region us-east-2
```

출력:

```
{
```

```
    "TransitGatewayPeeringAttachment": {
        "TransitGatewayAttachmentId": "tgw-attach-4455667788aabbccd",
        "RequesterTgwInfo": {
            "TransitGatewayId": "tgw-123abc05e04123abc",
            "OwnerId": "123456789012",
            "Region": "us-west-2"
        },
        "AccepterTgwInfo": {
            "TransitGatewayId": "tgw-11223344aabbcc112",
            "OwnerId": "123456789012",
            "Region": "us-east-2"
        },
        "State": "pending",
        "CreationTime": "2019-12-09T11:38:31.000Z"
    }
}
```

자세한 내용은 Transit Gateways 가이드의 Transit Gateway 피어링 연결을 참조하세요.

- 자세한 API 내용은 명령 참조AcceptTransitGatewayPeeringAttachment의 섹션을 참조하세요. AWS CLI

### `accept-transit-gateway-vpc-attachment`

다음 코드 예시에서는 `accept-transit-gateway-vpc-attachment`을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이에 를 연결하라는 요청을 수락VPC합니다.

다음 `accept-transit-gateway-vpc-attachment` 예제에서는 지정된 연결에 대한 요청을 수락합니다.

```
aws ec2 accept-transit-gateway-vpc-attachment \
    --transit-gateway-attachment-id tgw-attach-0a34fe6b4fEXAMPLE
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 Transit Gateways 가이드의 Transit Gateway Attachments to VPC a를 참조하세요.

- 자세한 API 내용은 명령 참조AcceptTransitGatewayVpcAttachment의 섹션을 참조하세요. AWS CLI

### `accept-vpc-endpoint-connections`

다음 코드 예시에서는 `accept-vpc-endpoint-connections`을 사용하는 방법을 보여 줍니다.

AWS CLI

인터페이스 엔드포인트 연결 요청을 수락하려면

이 예제에서는 지정된 엔드포인트 서비스에 대해 지정된 엔드포인트 연결 요청을 수락합니다.

명령:

```
aws ec2 accept-vpc-endpoint-connections --service-id vpce-svc-03d5ebb7d9579a2b3 --
vpc-endpoint-ids vpce-0c1308d7312217abc
```

출력:

```
{
  "Unsuccessful": []
}
```

- 자세한 API 내용은 명령 참조AcceptVpcEndpointConnections의 섹션을 참조하세요. AWS CLI

### accept-vpc-peering-connection

다음 코드 예시에서는 accept-vpc-peering-connection을 사용하는 방법을 보여 줍니다.

AWS CLI

VPC 피어링 연결을 수락하려면

이 예제에서는 지정된 VPC 피어링 연결 요청을 수락합니다.

명령:

```
aws ec2 accept-vpc-peering-connection --vpc-peering-connection-id pcx-1a2b3c4d
```

출력:

```
{
  "VpcPeeringConnection": {
    "Status": {
      "Message": "Provisioning",
      "Code": "provisioning"
    },
    "Tags": [],
    "AccepterVpcInfo": {
      "OwnerId": "444455556666",
      "VpcId": "vpc-44455566",
      "CidrBlock": "10.0.1.0/28"
    },
    "VpcPeeringConnectionId": "pcx-1a2b3c4d",
    "RequesterVpcInfo": {
      "OwnerId": "444455556666",
      "VpcId": "vpc-111abc45",
      "CidrBlock": "10.0.0.0/28"
    }
  }
}
```

- 자세한 API 내용은 명령 참조AcceptVpcPeeringConnection의 섹션을 참조하세요. AWS CLI

### advertise-byoip-cidr

다음 코드 예시에서는 advertise-byoip-cidr을 사용하는 방법을 보여 줍니다.

AWS CLI

주소 범위를 광고하려면

다음 `advertise-byoip-cidr` 예제에서는 지정된 퍼블릭 IPv4 주소 범위를 광고합니다.

```
aws ec2 advertise-byoip-cidr \
    --cidr 203.0.113.25/24
```

출력:

```
{
    "ByoipCidr": {
        "Cidr": "203.0.113.25/24",
        "StatusMessage": "ipv4pool-ec2-1234567890abcdef0",
        "State": "provisioned"
    }
}
```

- 자세한 API 내용은 명령 참조AdvertiseByoipCidr의 섹션을 참조하세요. AWS CLI

### `allocate-address`

다음 코드 예시에서는 `allocate-address`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: Amazon 주소 풀에서 탄력적 IP 주소를 할당하는 방법

다음 `allocate-address` 예제는 탄력적 IP 주소를 할당합니다. Amazon은 Amazon의 주소 풀에서 주소를 EC2 선택합니다.

```
aws ec2 allocate-address
```

출력:

```
{
    "PublicIp": "70.224.234.241",
    "AllocationId": "eipalloc-01435ba59eEXAMPLE",
    "PublicIpv4Pool": "amazon",
```

```
    "NetworkBorderGroup": "us-west-2",
    "Domain": "vpc"
}
```

자세한 내용은 Amazon EC2 사용 설명서의 탄력적 IP 주소를 참조하세요.

예제 2: 탄력적 IP 주소를 할당하고 네트워크 경계 그룹에 연결하는 방법

다음 `allocate-address` 예제에서는 탄력적 IP 주소를 할당하고 해당 주소를 지정된 네트워크 경계 그룹에 연결합니다.

```
aws ec2 allocate-address \
    --network-border-group us-west-2-lax-1
```

출력:

```
{
    "PublicIp": "70.224.234.241",
    "AllocationId": "eipalloc-e03dd489ceEXAMPLE",
    "PublicIpv4Pool": "amazon",
    "NetworkBorderGroup": "us-west-2-lax-1",
    "Domain": "vpc"
}
```

자세한 내용은 Amazon EC2 사용 설명서의 탄력적 IP 주소를 참조하세요.

예 3: 소유한 주소 풀에서 탄력적 IP 주소를 할당하는 방법

다음 `allocate-address` 예제에서는 Amazon Web Services 계정으로 가져온 주소 풀에서 탄력적 IP 주소를 할당합니다. Amazon은 주소 풀에서 주소를 EC2 선택합니다.

```
aws ec2 allocate-address \
    --public-ipv4-pool ipv4pool-ec2-1234567890abcdef0
```

출력:

```
{
    "AllocationId": "eipalloc-02463d08ceEXAMPLE",
    "NetworkBorderGroup": "us-west-2",
    "CustomerOwnedIp": "18.218.95.81",
```

```
    "CustomerOwnedIpv4Pool": "ipv4pool-ec2-1234567890abcdef0",
    "Domain": "vpc"
    "NetworkBorderGroup": "us-west-2",
}
```

자세한 내용은 Amazon EC2 사용 설명서의 탄력적 IP 주소를 참조하세요.

- 자세한 API 내용은 명령 참조AllocateAddress의 섹션을 참조하세요. AWS CLI

### `allocate-hosts`

다음 코드 예시에서는 `allocate-hosts`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 전용 호스트 할당

다음 `allocate-hosts` 예제에서는 `m5.large` 인스턴스를 시작할 수 있는 `eu-west-1a` 가용 영역에 단일 전용 호스트를 할당합니다. 기본적으로 전용 호스트는 대상 인스턴스 시작만 허용하며 호스트 복구를 지원하지 않습니다.

```
aws ec2 allocate-hosts \
    --instance-type m5.large \
    --availability-zone eu-west-1a \
    --quantity 1
```

출력:

```
{
    "HostIds": [
        "h-07879acf49EXAMPLE"
    ]
}
```

예제 2: 자동 배치 및 호스트 복구가 활성화된 전용 호스트 할당

다음 `allocate-hosts` 예제에서는 자동 배치 및 호스트 복구가 활성화된 `eu-west-1a` 가용 영역에 단일 전용 호스트를 할당합니다.

```
aws ec2 allocate-hosts \
    --instance-type m5.large \
```

```
    --availability-zone eu-west-1a \
    --auto-placement on \
    --host-recovery on \
    --quantity 1
```

출력:

```
{
     "HostIds": [
         "h-07879acf49EXAMPLE"
     ]
}
```

예제 3: 태그가 있는 전용 호스트 할당

다음 `allocate-hosts` 예제에서는 단일 전용 호스트를 할당하고 이름이 `purpose` 이고 값이 인 태그를 적용합니다`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}'
```

출력:

```
{
    "HostIds": [
        "h-07879acf49EXAMPLE"
    ]
}
```

자세한 내용은 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 전용 호스트 할당을 참조하세요.

- 자세한 API 내용은 명령 참조AllocateHosts의 섹션을 참조하세요. AWS CLI

### `allocate-ipam-pool-cidr`

다음 코드 예시에서는 `allocate-ipam-pool-cidr`을 사용하는 방법을 보여 줍니다.

AWS CLI

IPAM 풀CIDR에서 를 할당하려면

다음 `allocate-ipam-pool-cidr` 예제에서는 IPAM 풀CIDR에서 를 할당합니다.

(Linux):

```
aws ec2 allocate-ipam-pool-cidr \
    --ipam-pool-id ipam-pool-0533048da7d823723 \
    --netmask-length 24
```

(Windows):

```
aws ec2 allocate-ipam-pool-cidr ^
   --ipam-pool-id ipam-pool-0533048da7d823723 ^
   --netmask-length 24
```

출력:

```
{
    "IpamPoolAllocation": {
        "Cidr": "10.0.0.0/24",
        "IpamPoolAllocationId": "ipam-pool-alloc-018ecc28043b54ba38e2cd99943cebfbd",
        "ResourceType": "custom",
        "ResourceOwner": "123456789012"
    }
}
```

자세한 내용은 Amazon VPC IPAM 사용 설명서의 IP 주소 공간을 예약하기 위해 CIDR 풀에 수동으로 할당을 참조하세요.

- 자세한 API 내용은 명령 참조AllocateIpamPoolCidr의 섹션을 참조하세요. AWS CLI

### apply-security-groups-to-client-vpn-target-network

다음 코드 예시에서는 `apply-security-groups-to-client-vpn-target-network`을 사용하는 방법을 보여 줍니다.

AWS CLI

클라이언트 VPN 엔드포인트의 대상 네트워크에 보안 그룹을 적용하려면

다음 apply-security-groups-to-client-vpn-target-network 예제에서는 지정된 대상 네트워크와 클라이언트 VPN 엔드포인트 간의 sg-01f6e627a89f4db32 연결에 보안 그룹을 적용합니다.

```
aws ec2 apply-security-groups-to-client-vpn-target-network \
    --security-group-ids sg-01f6e627a89f4db32 \
    --vpc-id vpc-0e2110c2f324332e0 \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

출력:

```
{
    "SecurityGroupIds": [
        "sg-01f6e627a89f4db32"
    ]
}
```

자세한 내용은 AWS 클라이언트 VPN 관리자 안내서의 대상 네트워크를 참조하세요.

- 자세한 API 내용은 명령 참조ApplySecurityGroupsToClientVpnTargetNetwork의 섹션을 참조하세요. AWS CLI

### assign-ipv6-addresses

다음 코드 예시에서는 assign-ipv6-addresses을 사용하는 방법을 보여 줍니다.

AWS CLI

네트워크 인터페이스에 특정 IPv6 주소를 할당하려면

이 예제에서는 지정된 IPv6 주소를 지정된 네트워크 인터페이스에 할당합니다.

명령:

```
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
```

출력:

```
{
  "AssignedIpv6Addresses": [
```

```
        "2001:db8:1234:1a00:3304:8879:34cf:4071",
        "2001:db8:1234:1a00:9691:9503:25ad:1761"
    ],
    "NetworkInterfaceId": "eni-38664473"
}
```

Amazon이 선택한 IPv6 주소를 네트워크 인터페이스에 할당하려면

이 예제에서는 지정된 네트워크 인터페이스에 두 개의 IPv6 주소를 할당합니다. Amazon은 서브넷의 IPv6 CIDR 블록 범위에 있는 사용 가능한 IPv6 주소에서 이러한 IPv6 주소를 자동으로 할당합니다.

명령:

```
aws ec2 assign-ipv6-addresses --network-interface-id eni-38664473 --ipv6-address-
count 2
```

출력:

```
{
    "AssignedIpv6Addresses": [
        "2001:db8:1234:1a00:3304:8879:34cf:4071",
        "2001:db8:1234:1a00:9691:9503:25ad:1761"
    ],
    "NetworkInterfaceId": "eni-38664473"
}
```

- API 자세한 내용은 AWS CLI 명령 참조의 AssignIpv6Addresses하세요.

### assign-private-ip-addresses

다음 코드 예시에서는 assign-private-ip-addresses을 사용하는 방법을 보여 줍니다.

AWS CLI

네트워크 인터페이스에 특정 보조 프라이빗 IP 주소를 할당하려면

이 예제에서는 지정된 보조 프라이빗 IP 주소를 지정된 네트워크 인터페이스에 할당합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 assign-private-ip-addresses --network-interface-id eni-e5aa89a3 --private-
ip-addresses 10.0.0.82
```

Amazon이 EC2 선택한 보조 프라이빗 IP 주소를 네트워크 인터페이스에 할당하려면

이 예제에서는 지정된 네트워크 인터페이스에 두 개의 보조 프라이빗 IP 주소를 할당합니다. Amazon은 네트워크 인터페이스가 연결된 서브넷의 CIDR 블록 범위에 있는 사용 가능한 IP 주소에서 이러한 IP 주소를 EC2 자동으로 할당합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 assign-private-ip-addresses --network-interface-id eni-e5aa89a3 --secondary-
private-ip-address-count 2
```

- 자세한 API 내용은 명령 참조AssignPrivateIpAddresses의 섹션을 참조하세요. AWS CLI

### assign-private-nat-gateway-address

다음 코드 예시에서는 assign-private-nat-gateway-address을 사용하는 방법을 보여 줍니다.

AWS CLI

프라이빗 NAT 게이트웨이에 프라이빗 IP 주소를 할당하려면

다음 assign-private-nat-gateway-address 예제에서는 지정된 프라이빗 NAT 게이트웨이에 두 개의 프라이빗 IP 주소를 할당합니다.

```
aws ec2 assign-private-nat-gateway-address \
    --nat-gateway-id nat-1234567890abcdef0 \
    --private-ip-address-count 2
```

출력:

```
{
    "NatGatewayId": "nat-1234567890abcdef0",
    "NatGatewayAddresses": [
        {
            "NetworkInterfaceId": "eni-0065a61b324d1897a",
            "IsPrimary": false,
            "Status": "assigning"
        },
```

```
        {
            "NetworkInterfaceId": "eni-0065a61b324d1897a",
            "IsPrimary": false,
            "Status": "assigning"
        }
    ]
}
```

자세한 내용은 Amazon VPC 사용 설명서의 NAT 게이트웨이를 참조하세요.

- 자세한 API 내용은 명령 참조AssignPrivateNatGatewayAddress의 섹션을 참조하세요. AWS CLI

### associate-address

다음 코드 예시에서는 associate-address을 사용하는 방법을 보여 줍니다.

AWS CLI

EC2-Classic에서 탄력적 IP 주소를 연결하려면

이 예제에서는 탄력적 IP 주소를 EC2-Classic의 인스턴스와 연결합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 associate-address --instance-id i-07ffe74c7330ebf53 --public-ip 198.51.100.0
```

에서 탄력적 IP 주소를 연결하려면 EC2VPC

이 예제에서는 탄력적 IP 주소를 의 인스턴스와 연결합니다VPC.

명령:

```
aws ec2 associate-address --instance-id i-0b263919b6498b123 --allocation-
id eipalloc-64d5890a
```

출력:

```
{
    "AssociationId": "eipassoc-2bebb745"
}
```

이 예제에서는 네트워크 인터페이스에 탄력적 IP 주소를 연결합니다.

명령:

```
aws ec2 associate-address --allocation-id eipalloc-64d5890a --network-interface-
id eni-1a2b3c4d
```

이 예제에서는 네트워크 인터페이스와 연결된 프라이빗 IP 주소에 탄력적 IP를 연결합니다.

명령:

```
aws ec2 associate-address --allocation-id eipalloc-64d5890a --network-interface-
id eni-1a2b3c4d --private-ip-address 10.0.0.85
```

- 자세한 API 내용은 명령 참조AssociateAddress의 섹션을 참조하세요. AWS CLI

### associate-client-vpn-target-network

다음 코드 예시에서는 associate-client-vpn-target-network을 사용하는 방법을 보여 줍니다.

AWS CLI

대상 네트워크를 클라이언트 VPN 엔드포인트에 연결하려면

다음 associate-client-vpn-target-network 예제에서는 서브넷을 지정된 클라이언트 VPN 엔드포인트와 연결합니다.

```
aws ec2 associate-client-vpn-target-network \
    --subnet-id subnet-0123456789abcabca \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

출력:

```
{
    "AssociationId": "cvpn-assoc-12312312312312312",
    "Status": {
        "Code": "associating"
    }
}
```

자세한 내용은 AWS 클라이언트 VPN 관리자 안내서의 대상 네트워크를 참조하세요.

- 자세한 API 내용은 명령 참조AssociateClientVpnTargetNetwork의 섹션을 참조하세요. AWS CLI

### associate-dhcp-options

다음 코드 예시에서는 associate-dhcp-options을 사용하는 방법을 보여 줍니다.

AWS CLI

DHCP 옵션 세트를 에 연결하려면 VPC

이 예제에서는 지정된 DHCP 옵션 세트를 지정된 와 연결합니다VPC. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 associate-dhcp-options --dhcp-options-id dopt-d9070ebb --vpc-id vpc-a01106c2
```

기본 DHCP 옵션 세트를 에 연결하려면 VPC

이 예제에서는 기본 DHCP 옵션 세트를 지정된 와 연결합니다VPC. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 associate-dhcp-options --dhcp-options-id default --vpc-id vpc-a01106c2
```

- 자세한 API 내용은 명령 참조AssociateDhcpOptions의 섹션을 참조하세요. AWS CLI

### associate-iam-instance-profile

다음 코드 예시에서는 associate-iam-instance-profile을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 인스턴스 프로파일과 인스턴스를 연결하려면

이 예제에서는 라는 IAM 인스턴스 프로파일을 인스턴스 admin-role와 연결합니다i-123456789abcde123.

명령:

```
aws ec2 associate-iam-instance-profile --instance-id i-123456789abcde123 --iam-
instance-profile Name=admin-role
```

출력:

```
{
  "IamInstanceProfileAssociation": {
      "InstanceId": "i-123456789abcde123",
      "State": "associating",
      "AssociationId": "iip-assoc-0e7736511a163c209",
      "IamInstanceProfile": {
          "Id": "AIPAJBLK7RKJKWDXVHIEC",
          "Arn": "arn:aws:iam::123456789012:instance-profile/admin-role"
      }
  }
}
```

- 자세한 API 내용은 명령 참조AssociateIamInstanceProfile의 섹션을 참조하세요. AWS CLI

### associate-instance-event-window

다음 코드 예시에서는 associate-instance-event-window을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 하나 이상의 인스턴스를 이벤트 창에 연결하려면

다음 associate-instance-event-window 예제에서는 하나 이상의 인스턴스를 이벤트 창과 연결합니다.

```
aws ec2 associate-instance-event-window \
    --region us-east-1 \
    --instance-event-window-id iew-0abcdef1234567890 \
    --association-target "InstanceIds=i-1234567890abcdef0,i-0598c7d356eba48d7"
```

출력:

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
```

```
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [
                "i-1234567890abcdef0",
                "i-0598c7d356eba48d7"
            ],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating"
    }
}
```

이벤트 기간 제약 조건은 Amazon EC2 사용 설명서의 예약된 이벤트 섹션의 고려 사항을 참조하세요.

예제 2: 인스턴스 태그를 이벤트 창과 연결하려면

다음 `associate-instance-event-window` 예제에서는 인스턴스 태그를 이벤트 창과 연결합니다. `instance-event-window-id` 파라미터를 입력하여 이벤트 창을 지정합니다. 인스턴스 태그를 연결하려면 `association-target` 파라미터를 지정하고 파라미터 값에는 하나 이상의 태그를 지정합니다.

```
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}]"
```

출력:

```
{
    "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"
    }
}
```

이벤트 기간 제약 조건은 Amazon EC2 사용 설명서의 예약된 이벤트 섹션의 고려 사항을 참조하세요.

예제 3: 전용 호스트를 이벤트 창에 연결하려면

다음 `associate-instance-event-window` 예제에서는 전용 호스트를 이벤트 창과 연결합니다. `instance-event-window-id` 파라미터를 입력하여 이벤트 창을 지정합니다. 전용 호스트를 연결하려면 `--association-target` 파라미터를 지정하고 파라미터 값에는 하나 이상의 전용 호스트 를 지정합니다IDs.

```
aws ec2 associate-instance-event-window \
    --region us-east-1 \
    --instance-event-window-id iew-0abcdef1234567890 \
    --association-target "DedicatedHostIds=h-029fa35a02b99801d"
```

출력:

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [],
            "Tags": [],
            "DedicatedHostIds": [
                "h-029fa35a02b99801d"
            ]
        },
        "State": "creating"
    }
```

```
}
```

이벤트 기간 제약 조건은 Amazon EC2 사용 설명서의 예약된 이벤트 섹션의 고려 사항을 참조하세요.

- 자세한 API 내용은 명령 참조AssociateInstanceEventWindow의 섹션을 참조하세요. AWS CLI

### associate-ipam-resource-discovery

다음 코드 예시에서는 associate-ipam-resource-discovery을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 검색을 에 연결하려면 IPAM

이 예제에서는 IPAM 위임된 관리자이고 다른 AWS 계정에서 리소스 검색을 생성하여 공유했으므로 IPAM를 사용하여 다른 계정에서 CIDRs 소유한 리소스를 관리하고 모니터링할 수 있습니다.

참고

이 요청을 완료하려면 에서 가져올 수 있는 리소스 검색 IDdescribe-ipam-resource-discoveries와 describe-ipams 에서 가져올 수 있는 IPAM ID가 필요합니다. 연결하려는 리소스 검색은 먼저 를 사용하여 계정과 공유되어야 합니다 AWS RAM. --region 입력한 는 연결IPAM하려는 의 홈 리전과 일치해야 합니다.

다음 associate-ipam-resource-discovery 예제에서는 리소스 검색을 와 연결합니다IPAM.

```
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
```

출력:

```
{
    {
        "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": []
        }
    }
}
```

리소스 검색을 연결하면 다른 계정에서 생성한 리소스의 IP 주소를 모니터링 및/또는 관리할 수 있습니다. 자세한 내용은 Amazon VPC IPAM 사용 설명서의 조직 외부 계정IPAM과 통합을 참조하세요.

- 자세한 API 내용은 명령 참조AssociateIpamResourceDiscovery의 섹션을 참조하세요. AWS CLI

### associate-nat-gateway-address

다음 코드 예시에서는 associate-nat-gateway-address을 사용하는 방법을 보여 줍니다.

AWS CLI

탄력적 IP 주소를 퍼블릭 NAT 게이트웨이에 연결하려면

다음 associate-nat-gateway-address 예제에서는 지정된 탄력적 IP 주소를 지정된 퍼블릭 NAT 게이트웨이와 연결합니다. 는 보조 프라이빗 IPv4 주소를 AWS 자동으로 할당합니다.

```
aws ec2 associate-nat-gateway-address \
    --nat-gateway-id nat-1234567890abcdef0 \
    --allocation-ids eipalloc-0be6ecac95EXAMPLE
```

출력:

```
{
    "NatGatewayId": "nat-1234567890abcdef0",
    "NatGatewayAddresses": [
        {
            "AllocationId": "eipalloc-0be6ecac95EXAMPLE",
```

```
            "NetworkInterfaceId": "eni-09cc4b2558794f7f9",
            "IsPrimary": false,
            "Status": "associating"
        }
    ]
}
```

자세한 내용은 Amazon VPC 사용 설명서의 NAT 게이트웨이를 참조하세요.

- 자세한 API 내용은 명령 참조AssociateNatGatewayAddress의 섹션을 참조하세요. AWS CLI

### associate-route-table

다음 코드 예시에서는 associate-route-table을 사용하는 방법을 보여 줍니다.

AWS CLI

라우팅 테이블을 서브넷에 연결하려면

이 예제에서는 지정된 라우팅 테이블을 지정된 서브넷과 연결합니다.

명령:

```
aws ec2 associate-route-table --route-table-id rtb-22574640 --subnet-
id subnet-9d4a7b6c
```

출력:

```
{
    "AssociationId": "rtbassoc-781d0d1a"
}
```

- 자세한 API 내용은 명령 참조AssociateRouteTable의 섹션을 참조하세요. AWS CLI

### associate-subnet-cidr-block

다음 코드 예시에서는 associate-subnet-cidr-block을 사용하는 방법을 보여 줍니다.

AWS CLI

IPv6 CIDR 블록을 서브넷에 연결하려면

이 예제에서는 IPv6 CIDR 블록을 지정된 서브넷과 연결합니다.

명령:

```
aws ec2 associate-subnet-cidr-block --subnet-id subnet-5f46ec3b --ipv6-cidr-
block 2001:db8:1234:1a00::/64
```

출력:

```
{
  "SubnetId": "subnet-5f46ec3b",
  "Ipv6CidrBlockAssociation": {
      "Ipv6CidrBlock": "2001:db8:1234:1a00::/64",
      "AssociationId": "subnet-cidr-assoc-3aa54053",
      "Ipv6CidrBlockState": {
          "State": "associating"
      }
  }
}
```

- 자세한 API 내용은 명령 참조AssociateSubnetCidrBlock의 섹션을 참조하세요. AWS CLI

### associate-transit-gateway-multicast-domain

다음 코드 예시에서는 associate-transit-gateway-multicast-domain을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이를 멀티캐스트 도메인에 연결하려면

다음 associate-transit-gateway-multicast-domain 예제에서는 지정된 서브넷 및 연결을 지정된 멀티캐스트 도메인과 연결합니다.

```
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
```

출력:

```
{
    "Associations": {
        "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef79d6e597",
        "TransitGatewayAttachmentId": "tgw-attach-028c1dd0f8f5cbe8e",
        "ResourceId": "vpc-01128d2c240c09bd5",
        "ResourceType": "vpc",
        "Subnets": [
            {
                "SubnetId": "subnet-000de86e3b49c932a",
                "State": "associating"
            }
        ]
    }
}
```

자세한 내용은 Transit Gateways 가이드의 멀티캐스트 도메인 관리를 참조하세요.

- 자세한 API 내용은 명령 참조AssociateTransitGatewayMulticastDomain의 섹션을 참조하세요. AWS CLI

### associate-transit-gateway-route-table

다음 코드 예시에서는 associate-transit-gateway-route-table을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 라우팅 테이블을 전송 게이트웨이 연결에 연결하려면

다음 예제에서는 지정된 전송 게이트웨이 라우팅 테이블을 지정된 VPC 연결과 연결합니다.

```
aws ec2 associate-transit-gateway-route-table \
    --transit-gateway-route-table-id tgw-rtb-002573ed1eEXAMPLE \
    --transit-gateway-attachment-id tgw-attach-0b5968d3b6EXAMPLE
```

출력:

```
{
    "Association": {
        "TransitGatewayRouteTableId": "tgw-rtb-002573ed1eEXAMPLE",
        "TransitGatewayAttachmentId": "tgw-attach-0b5968d3b6EXAMPLE",
        "ResourceId": "vpc-0065acced4EXAMPLE",
```

```
        "ResourceType": "vpc",
        "State": "associating"
    }
}
```

자세한 내용은 Transit Gateways 가이드의 Transit Gateway Route Table 연결을 AWS 참조하세요.

- 자세한 API 내용은 명령 참조AssociateTransitGatewayRouteTable의 섹션을 참조하세요. AWS CLI

### associate-vpc-cidr-block

다음 코드 예시에서는 associate-vpc-cidr-block을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: Amazon 제공 IPv6 CIDR 블록을 와 연결하려면 VPC

다음 associate-vpc-cidr-block 예제는 IPv6 CIDR 블록을 지정된 와 연결합니다VPC.

```
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
```

출력:

```
{
    "Ipv6CidrBlockAssociation": {
        "AssociationId": "vpc-cidr-assoc-0838ce7d9dEXAMPLE",
        "Ipv6CidrBlockState": {
            "State": "associating"
        },
        "NetworkBorderGroup": "us-west-2-lax-1"
    },
    "VpcId": "vpc-8EXAMPLE"
}
```

예제 2: 추가 IPv4 CIDR 블록을 와 연결하려면 VPC

다음 associate-vpc-cidr-block 예제는 IPv4 CIDR 블록을 지정된 10.2.0.0/16와 연결합니다VPC.

```
aws ec2 associate-vpc-cidr-block \
    --vpc-id vpc-1EXAMPLE \
    --cidr-block 10.2.0.0/16
```

출력:

```
{
    "CidrBlockAssociation": {
        "AssociationId": "vpc-cidr-assoc-2EXAMPLE",
        "CidrBlock": "10.2.0.0/16",
        "CidrBlockState": {
            "State": "associating"
        }
    },
    "VpcId": "vpc-1EXAMPLE"
}
```

- 자세한 API 내용은 명령 참조AssociateVpcCidrBlock의 섹션을 참조하세요. AWS CLI

### attach-classic-link-vpc

다음 코드 예시에서는 attach-classic-link-vpc을 사용하는 방법을 보여 줍니다.

AWS CLI

EC2-Classic 인스턴스를 에 연결(연결)하려면 VPC

이 예제에서는 VPC 보안 그룹 sg-1234567890를 통해 인스턴스 i-88888888abcdef0을 VPC vpc-12312312에 연결합니다.

명령:

```
aws ec2 attach-classic-link-vpc --instance-id  i-1234567890abcdef0 --vpc-
id vpc-88888888 --groups sg-12312312
```

출력:

```
{
  "Return": true
}
```

- 자세한 API 내용은 명령 참조AttachClassicLinkVpc의 섹션을 참조하세요. AWS CLI

### attach-internet-gateway

다음 코드 예시에서는 attach-internet-gateway을 사용하는 방법을 보여 줍니다.

AWS CLI

에 인터넷 게이트웨이를 연결하려면 VPC

다음 attach-internet-gateway 예제에서는 지정된 인터넷 게이트웨이를 특정 에 연결합니다 VPC.

```
aws ec2 attach-internet-gateway \
    --internet-gateway-id igw-0d0fb496b3EXAMPLE \
    --vpc-id vpc-0a60eb65b4EXAMPLE
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon VPC 사용 설명서의 인터넷 게이트웨이를 참조하세요.

- 자세한 API 내용은 명령 참조AttachInternetGateway의 섹션을 참조하세요. AWS CLI

### attach-network-interface

다음 코드 예시에서는 attach-network-interface을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 인스턴스에 네트워크 인터페이스 연결

다음 attach-network-interface 예제에서는 지정된 네트워크 인터페이스를 지정된 인스턴스에 연결합니다.

```
aws ec2 attach-network-interface \
    --network-interface-id eni-0dc56a8d4640ad10a \
    --instance-id i-1234567890abcdef0 \
    --device-index 1
```

출력:

```
{
    "AttachmentId": "eni-attach-01a8fc87363f07cf9"
}
```

자세한 내용은 Amazon EC2 사용 설명서의 탄력적 네트워크 인터페이스를 참조하세요.

예제 2: 여러 네트워크 카드가 있는 인스턴스에 네트워크 인터페이스를 연결하려면

다음 `attach-network-interface` 예제에서는 지정된 네트워크 인터페이스를 지정된 인스턴스 및 네트워크 카드에 연결합니다.

```
aws ec2 attach-network-interface \
    --network-interface-id eni-07483b1897541ad83 \
    --instance-id i-01234567890abcdef \
    --network-card-index 1 \
    --device-index 1
```

출력:

```
{
    "AttachmentId": "eni-attach-0fbd7ee87a88cd06c"
}
```

자세한 내용은 Amazon EC2 사용 설명서의 탄력적 네트워크 인터페이스를 참조하세요.

- 자세한 API 내용은 명령 참조AttachNetworkInterface의 섹션을 참조하세요. AWS CLI

### attach-verified-access-trust-provider

다음 코드 예시에서는 `attach-verified-access-trust-provider`을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스에 신뢰 공급자를 연결하려면

다음 `attach-verified-access-trust-provider` 예제에서는 지정된 Verified Access 신뢰 공급자를 지정된 Verified Access 인스턴스에 연결합니다.

```
aws ec2 attach-verified-access-trust-provider \
    --verified-access-instance-id vai-0ce000c0b7643abea \
```

```
    --verified-access-trust-provider-id vatp-0bb32de759a3e19e7
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Verified Access 사용 설명서의 Verified Access 인스턴스를 참조하세요. AWS

- 자세한 API 내용은 명령 참조AttachVerifiedAccessTrustProvider의 섹션을 참조하세요. AWS CLI

### `attach-volume`

다음 코드 예시에서는 `attach-volume`을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스에 볼륨을 연결하려면

이 예제 명령은 볼륨(`vol-1234567890abcdef0`)을 인스턴스(`i-01474ef662b89480`)에 로 연결합니다`/dev/sdf`.

명령:

```
aws ec2 attach-volume --volume-id vol-1234567890abcdef0 --instance-
id i-01474ef662b89480 --device /dev/sdf
```

출력:

```
{
    "AttachTime": "YYYY-MM-DDTHH:MM:SS.000Z",
    "InstanceId": "i-01474ef662b89480",
    "VolumeId": "vol-1234567890abcdef0",
    "State": "attaching",
    "Device": "/dev/sdf"
}
```

- 자세한 API 내용은 명령 참조AttachVolume의 섹션을 참조하세요. AWS CLI

### attach-vpn-gateway

다음 코드 예시에서는 attach-vpn-gateway을 사용하는 방법을 보여 줍니다.

AWS CLI

에 가상 프라이빗 게이트웨이를 연결하려면 VPC

다음 attach-vpn-gateway 예제에서는 지정된 가상 프라이빗 게이트웨이를 지정된 에 연결합니다VPC.

```
aws ec2 attach-vpn-gateway \
    --vpn-gateway-id vgw-9a4cacf3 \
    --vpc-id vpc-a01106c2
```

출력:

```
{
    "VpcAttachment": {
        "State": "attaching",
        "VpcId": "vpc-a01106c2"
    }
}
```

- 자세한 API 내용은 명령 참조AttachVpnGateway의 섹션을 참조하세요. AWS CLI

### authorize-client-vpn-ingress

다음 코드 예시에서는 authorize-client-vpn-ingress을 사용하는 방법을 보여 줍니다.

AWS CLI

클라이언트 VPN 엔드포인트에 대한 권한 부여 규칙을 추가하려면

다음 authorize-client-vpn-ingress 예제에서는 모든 클라이언트가 인터넷()에 액세스할 수 있도록 허용하는 수신 권한 부여 규칙을 추가합니다0.0.0.0/0.

```
aws ec2 authorize-client-vpn-ingress \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \
    --target-network-cidr 0.0.0.0/0 \
    --authorize-all-groups
```

출력:

```
{
    "Status": {
        "Code": "authorizing"
    }
}
```

자세한 내용은 AWS 클라이언트 VPN 관리자 안내서의 권한 부여 규칙을 참조하세요.

- 자세한 API 내용은 명령 참조AuthorizeClientVpnIngress의 섹션을 참조하세요. AWS CLI

### authorize-security-group-egress

다음 코드 예시에서는 authorize-security-group-egress을 사용하는 방법을 보여 줍니다.

AWS CLI

아웃바운드 트래픽을 특정 주소 범위에 허용하는 규칙을 추가하려면

이 예제 명령은 TCP 포트 80의 지정된 주소 범위에 대한 액세스 권한을 부여하는 규칙을 추가합니다.

명령(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}]'
```

명령(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}]
```

특정 보안 그룹에 아웃바운드 트래픽을 허용하는 규칙을 추가하려면

이 예제 명령은 TCP 포트 80에서 지정된 보안 그룹에 대한 액세스 권한을 부여하는 규칙을 추가합니다.

명령(Linux):

```
aws ec2 authorize-security-group-egress --group-id sg-1a2b3c4d --ip-permissions
 IpProtocol=tcp,FromPort=80,ToPort=80,UserIdGroupPairs='[{GroupId=sg-4b51a32f}]'
```

명령(Windows):

```
aws ec2 authorize-security-group-egress --group-id sg-1a2b3c4d --ip-
permissions IpProtocol=tcp,FromPort=80,ToPort=80,UserIdGroupPairs=[{GroupId=sg-4b51a32f}]
```

- 자세한 API 내용은 명령 참조AuthorizeSecurityGroupEgress의 섹션을 참조하세요. AWS CLI

### authorize-security-group-ingress

다음 코드 예시에서는 authorize-security-group-ingress을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 인바운드 SSH 트래픽을 허용하는 규칙을 추가하려면

다음 authorize-security-group-ingress 예제에서는 TCP 포트 22()에서 인바운드 트래픽을 허용하는 규칙을 추가합니다SSH.

```
aws ec2 authorize-security-group-ingress \
    --group-id sg-1234567890abcdef0 \
```

```
    --protocol tcp \
    --port 22 \
    --cidr 203.0.113.0/24
```

출력:

```
{
    "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"
        }
    ]
}
```

예제 2: 다른 보안 그룹의 인바운드 HTTP 트래픽을 허용하는 규칙을 추가하려면

다음 `authorize-security-group-ingress` 예제에서는 소스 보안 그룹 에서 TCP 포트 80에 대한 인바운드 액세스를 허용하는 규칙을 추가합니다`sg-1a2b3c4d`. 소스 그룹은 동일하거나 VPC 피어에 있어야 합니다VPC(VPC피어링 연결 필요). 유입 트래픽은 퍼블릭 IP 주소 또는 탄력적 IP 주소가 아닌 소스 보안 그룹과 연결된 인스턴스의 프라이빗 IP 주소를 기반으로 허용됩니다.

```
aws ec2 authorize-security-group-ingress \
    --group-id sg-1234567890abcdef0 \
    --protocol tcp \
    --port 80 \
    --source-group sg-1a2b3c4d
```

출력:

```
{
    "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"
            }
        }
    ]
}
```

예제 3: 동일한 직접 호출에서 여러 규칙을 추가하는 방법

다음 `authorize-security-group-ingress` 예제에서는 `ip-permissions` 파라미터를 사용하여 TCP 포트 3389(RDP)에서 인바운드 액세스를 활성화하는 규칙과 ping/를 활성화하는 규칙의 두 가지 인바운드 규칙을 추가합니다ICMP.

aws ec2 authorize-security-group-ingress --group-id sg-1234567890abcdef0 --ip-permissions IpProtocol=tcp,FromPort=3389,ToPort=3389,IpRanges="[{CidrIp=172.31.0.0/16}]' IpProtocol'=icmp,FromPort=-1,ToPort=-1,IpRanges='[{CidrIp=172.31.0.0/16}]"

출력:

```
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-00e06e5d3690f29f3",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "123456789012",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 3389,
            "ToPort": 3389,
            "CidrIpv4": "172.31.0.0/16"
        },
        {
            "SecurityGroupRuleId": "sgr-0a133dd4493944b87",
            "GroupId": "sg-1234567890abcdef0",
```

```
            "GroupOwnerId": "123456789012",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": -1,
            "ToPort": -1,
            "CidrIpv4": "172.31.0.0/16"
        }
    ]
}
```

예제 4: ICMP 트래픽에 대한 규칙을 추가하려면

다음 `authorize-security-group-ingress` 예제에서는 `ip-permissions` 파라미터를 사용하여 어디서나 ICMP 메시지`Destination Unreachable: Fragmentation Needed and Don't Fragment was Set`(유형 3, 코드 4)를 허용하는 인바운드 규칙을 추가합니다.

```
aws ec2 authorize-security-group-ingress --group-id sg-1234567890abcdef0 --ip-permissions IpProtocol=icmp,FromPort=3,ToPort=4,IpRanges="[{CidrIp=0.0.0.0/0}]'
```

출력:

```
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-0de3811019069b787",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "123456789012",
            "IsEgress": false,
            "IpProtocol": "icmp",
            "FromPort": 3,
            "ToPort": 4,
            "CidrIpv4": "0.0.0.0/0"
        }
    ]
}
```

예제 5: IPv6 트래픽에 대한 규칙 추가

다음 `authorize-security-group-ingress` 예제에서는 `ip-permissions` 파라미터를 사용하여 IPv6 범위 에서 SSH 액세스(포트 22)를 허용하는 인바운드 규칙을 추가합니다`2001:db8:1234:1a00::/64`.

aws ec2 authorize-security-group-ingress --group-id sg-1234567890abcdef0 --ip-permissions IpProtocol=tcp,FromPort=22,ToPort=22,Ipv6Ranges ='[{CidrIpv6=2001:db8:1234:1a00::/64}]'

출력:

```
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-0455bc68b60805563",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "123456789012",
            "IsEgress": false,
            "IpProtocol": "tcp",
            "FromPort": 22,
            "ToPort": 22,
            "CidrIpv6": "2001:db8:1234:1a00::/64"
        }
    ]
}
```

예제 6: ICMPv6 트래픽에 대한 규칙을 추가하려면

다음 `authorize-security-group-ingress` 예제에서는 `ip-permissions` 파라미터를 사용하여 어디서나 ICMPv6 트래픽을 허용하는 인바운드 규칙을 추가합니다.

aws ec2 authorize-security-group-ingress --group-id sg-1234567890abcdef0 --ip-permissions IpProtocol=icmpv6,Ipv6Ranges ="[{CidrIpv6=::/0}]'

출력:

```
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-04b612d9363ab6327",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "123456789012",
            "IsEgress": false,
            "IpProtocol": "icmpv6",
            "FromPort": -1,
            "ToPort": -1,
            "CidrIpv6": "::/0"
```

```
        }
    ]
}
```

예제 7: 설명이 포함된 규칙 추가

다음 `authorize-security-group-ingress` 예제에서는 `ip-permissions` 파라미터를 사용하여 지정된 IPv4 주소 범위의 RDP 트래픽을 허용하는 인바운드 규칙을 추가합니다. 이 규칙에는 나중에 식별하는 데 도움이 되는 설명이 포함됩니다.

aws ec2 authorize-security-group-ingress --group-id sg-1234567890abcdef0 --ip-permissions IpProtocol=tcp,FromPort=3389,ToPort=3389,IpRanges='[{CidrIp=203.0.113.0/24,Description='RDP NY office'}]'

출력:

```
{
    "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"
        }
    ]
}
```

예제 8: 접두사 목록을 사용하는 인바운드 규칙을 추가하는 방법

다음 `authorize-security-group-ingress` 예제에서는 `ip-permissions` 파라미터를 사용하여 지정된 접두사 목록의 CIDR 범위에 대한 모든 트래픽을 허용하는 인바운드 규칙을 추가합니다.

aws ec2 authorize-security-group-ingress --group-id sg-04a351bfe432d4e71 --ip-permissions IpProtocol=all,PrefixListIds="[{PrefixListId=pl-002dc3ec097de1514}]'

출력:

```
{
    "Return": true,
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-09c74b32f677c6c7c",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "123456789012",
            "IsEgress": false,
            "IpProtocol": "-1",
            "FromPort": -1,
            "ToPort": -1,
            "PrefixListId": "pl-0721453c7ac4ec009"
        }
    ]
}
```

자세한 내용은 Amazon VPC 사용 설명서의 보안 그룹을 참조하세요.

- 자세한 API 내용은 명령 참조AuthorizeSecurityGroupIngress의 섹션을 참조하세요. AWS CLI

### bundle-instance

다음 코드 예시에서는 bundle-instance을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스를 번들링하려면

이 예제에서는 인스턴스를 라는 버킷i-1234567890abcdef0에 번들링합니다bundletasks. 액세스 키 의 값을 지정하기 전에 AWS 액세스 키 관리 모범 사례의 지침을 IDs검토하고 따릅니다.

명령:

```
aws ec2 bundle-instance --instance-id i-1234567890abcdef0 --bucket bundletasks --
prefix winami --owner-akid AK12AJEXAMPLE --owner-sak example123example
```

출력:

```
{
```

```
  "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"
  }
}
```

- 자세한 API 내용은 명령 참조BundleInstance의 섹션을 참조하세요. AWS CLI

### `cancel-bundle-task`

다음 코드 예시에서는 `cancel-bundle-task`을 사용하는 방법을 보여 줍니다.

AWS CLI

번들 작업을 취소하려면

이 예제에서는 번들 작업 을 취소합니다`bun-2a4e041c`.

명령:

```
aws ec2 cancel-bundle-task --bundle-id bun-2a4e041c
```

출력:

```
{
  "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"
  }
}
```

- 자세한 API 내용은 명령 참조CancelBundleTask의 섹션을 참조하세요. AWS CLI

### cancel-capacity-reservation-fleets

다음 코드 예시에서는 cancel-capacity-reservation-fleets을 사용하는 방법을 보여 줍니다.

AWS CLI

용량 예약 플릿을 취소하려면

다음 cancel-capacity-reservation-fleets 예제에서는 지정된 용량 예약 플릿과 예약 용량을 취소합니다. 플릿을 취소하면 플릿의 상태가 로 변경되며 cancelled더 이상 새 용량 예약을 생성할 수 없습니다. 또한 플릿의 개별 용량 예약이 모두 취소되고 이전에 예약된 용량으로 실행된 인스턴스는 공유 용량으로 계속 정상적으로 실행됩니다.

```
aws ec2 cancel-capacity-reservation-fleets \
    --capacity-reservation-fleet-ids crf-abcdef01234567890
```

출력:

```
{
    "SuccessfulFleetCancellations": [
        {
            "CurrentFleetState": "cancelling",
            "PreviousFleetState": "active",
            "CapacityReservationFleetId": "crf-abcdef01234567890"
        }
    ],
    "FailedFleetCancellations": []
}
```

용량 예약 플릿에 대한 자세한 내용은 Amazon EC2 사용 설명서의 용량 예약 플릿을 참조하세요.

- 자세한 API 내용은 명령 참조CancelCapacityReservationFleets의 섹션을 참조하세요. AWS CLI

### cancel-capacity-reservation

다음 코드 예시에서는 cancel-capacity-reservation을 사용하는 방법을 보여 줍니다.

AWS CLI

용량 예약을 취소하려면

다음 cancel-capacity-reservation 예제에서는 지정된 용량 예약을 취소합니다.

```
aws ec2 cancel-capacity-reservation \
    --capacity-reservation-id cr-1234abcd56EXAMPLE
```

출력:

```
{
    "Return": true
}
```

자세한 내용은 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 용량 예약 취소를 참조하세요.

- 자세한 API 내용은 명령 참조CancelCapacityReservation의 섹션을 참조하세요. AWS CLI

### cancel-conversion-task

다음 코드 예시에서는 cancel-conversion-task을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스 또는 볼륨의 활성 변환을 취소하려면

이 예제에서는 태스크 ID import-i-fh95npoc와 연결된 업로드를 취소합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 cancel-conversion-task --conversion-task-id import-i-fh95npoc
```

- 자세한 API 내용은 명령 참조CancelConversionTask의 섹션을 참조하세요. AWS CLI

### cancel-export-task

다음 코드 예시에서는 cancel-export-task을 사용하는 방법을 보여 줍니다.

AWS CLI

활성 내보내기 작업을 취소하려면

이 예제에서는 작업 ID export-i-fgelt0i7을 사용하여 활성 내보내기 작업을 취소합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 cancel-export-task --export-task-id export-i-fgelt0i7
```

- 자세한 API 내용은 명령 참조CancelExportTask의 섹션을 참조하세요. AWS CLI

### cancel-image-launch-permission

다음 코드 예시에서는 cancel-image-launch-permission을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon Web Services 계정과 AMI 공유를 취소하려면

다음 cancel-image-launch-permission 예제에서는 지정된 의 AMI시작 권한에서 계정을 제거합니다.

```
aws ec2 cancel-image-launch-permission \
    --image-id ami-0123456789example \
    --region us-east-1
```

출력:

```
{
    "Return": true
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon Web Services 계정과 AMI 공유된 의 취소를 참조하세요. EC2

- 자세한 API 내용은 명령 참조CancelImageLaunchPermission의 섹션을 참조하세요. AWS CLI

### cancel-import-task

다음 코드 예시에서는 cancel-import-task을 사용하는 방법을 보여 줍니다.

AWS CLI

가져오기 작업을 취소하려면

다음 cancel-import-task 예제에서는 지정된 이미지 가져오기 작업을 취소합니다.

```
aws ec2 cancel-import-task \
    --import-task-id import-ami-1234567890abcdef0
```

출력:

```
{
    "ImportTaskId": "import-ami-1234567890abcdef0",
    "PreviousState": "active",
    "State": "deleting"
}
```

- 자세한 API 내용은 명령 참조CancelImportTask의 섹션을 참조하세요. AWS CLI

### cancel-reserved-instances-listing

다음 코드 예시에서는 cancel-reserved-instances-listing을 사용하는 방법을 보여 줍니다.

AWS CLI

예약 인스턴스 목록을 취소하려면

다음 cancel-reserved-instances-listing 예제에서는 지정된 예약 인스턴스 목록을 취소합니다.

```
aws ec2 cancel-reserved-instances-listing \
    --reserved-instances-listing-id 5ec28771-05ff-4b9b-aa31-9e57dexample
```

- 자세한 API 내용은 명령 참조CancelReservedInstancesListing의 섹션을 참조하세요. AWS CLI

### cancel-spot-fleet-requests

다음 코드 예시에서는 cancel-spot-fleet-requests을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 스팟 플릿 요청을 취소하고 연결된 인스턴스를 종료하려면

다음 cancel-spot-fleet-requests 예제에서는 스팟 플릿 요청을 취소하고 연결된 온디맨드 인스턴스 및 스팟 인스턴스를 종료합니다.

```
aws ec2 cancel-spot-fleet-requests \
    --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
    --terminate-instances
```

출력:

```
{
    "SuccessfulFleetRequests": [
        {
            "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
            "CurrentSpotFleetRequestState": "cancelled_terminating",
            "PreviousSpotFleetRequestState": "active"
        }
    ],
    "UnsuccessfulFleetRequests": []
}
```

자세한 내용은 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 스팟 플릿 요청 취소를 참조하세요.

예제 2: 연결된 인스턴스를 종료하지 않고 스팟 플릿 요청을 취소하려면

다음 cancel-spot-fleet-requests 예제에서는 연결된 온디맨드 인스턴스 및 스팟 인스턴스를 종료하지 않고 스팟 플릿 요청을 취소합니다.

```
aws ec2 cancel-spot-fleet-requests \
    --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE \
    --no-terminate-instances
```

출력:

```
{
    "SuccessfulFleetRequests": [
        {
            "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE",
            "CurrentSpotFleetRequestState": "cancelled_running",
            "PreviousSpotFleetRequestState": "active"
        }
    ],
    "UnsuccessfulFleetRequests": []
}
```

자세한 내용은 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 스팟 플릿 요청 취소를 참조하세요.

- 자세한 API 내용은 명령 참조CancelSpotFleetRequests의 섹션을 참조하세요. AWS CLI

### `cancel-spot-instance-requests`

다음 코드 예시에서는 `cancel-spot-instance-requests`을 사용하는 방법을 보여 줍니다.

AWS CLI

스팟 인스턴스 요청을 취소하려면

이 예제 명령은 스팟 인스턴스 요청을 취소합니다.

명령:

```
aws ec2 cancel-spot-instance-requests --spot-instance-request-ids sir-08b93456
```

출력:

```
{
    "CancelledSpotInstanceRequests": [
        {
            "State": "cancelled",
            "SpotInstanceRequestId": "sir-08b93456"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조CancelSpotInstanceRequests의 섹션을 참조하세요. AWS CLI

### confirm-product-instance

다음 코드 예시에서는 confirm-product-instance을 사용하는 방법을 보여 줍니다.

AWS CLI

제품 인스턴스를 확인하려면

이 예제에서는 지정된 제품 코드가 지정된 인스턴스와 연결되어 있는지 여부를 결정합니다.

명령:

```
aws ec2 confirm-product-instance --product-code 774F4FF8 --instance-
id i-1234567890abcdef0
```

출력:

```
{
  "OwnerId": "123456789012"
}
```

- 자세한 API 내용은 명령 참조ConfirmProductInstance의 섹션을 참조하세요. AWS CLI

### copy-fpga-image

다음 코드 예시에서는 copy-fpga-image을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon FPGA 이미지를 복사하려면

이 예제는 지정된 을 us-east-1 리전AFI에서 현재 리전()으로 복사합니다eu-west-1.

명령:

```
aws ec2 copy-fpga-image --name copy-afi --source-fpga-image-id afi-0d123e123bfc85abc
 --source-region us-east-1 --region eu-west-1
```

출력:

```
{
  "FpgaImageId": "afi-06b12350a123fbabc"
```

```
}
```

- 자세한 API 내용은 명령 참조CopyFpgaImage의 섹션을 참조하세요. AWS CLI

### copy-image

다음 코드 예시에서는 copy-image을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 를 다른 리전에 복사AMI하려면

다음 copy-image 예제 명령은 지정된 을 us-west-2 리전AMI에서 us-east-1 리전으로 복사하고 간단한 설명을 추가합니다.

```
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."
```

출력:

```
{
    "ImageId": "ami-0123456789abcdefg"
}
```

자세한 내용은 Amazon EC2 사용 설명서의 복사AMI를 참조하세요.

예제 2: 를 다른 리전AMI에 복사하고 백업 스냅샷을 암호화하는 방법

다음 copy-image 명령은 us-west-2 리전AMI에서 현재 리전으로 지정된 를 복사하고 지정된 KMS 키를 사용하여 백업 스냅샷을 암호화합니다.

```
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
```

출력:

```
{
    "ImageId": "ami-0123456789abcdefg"
}
```

자세한 내용은 Amazon EC2 사용 설명서의 복사AMI를 참조하세요.

예제 3: 를 복사할 때 사용자 정의 AMI 태그를 포함하려면 AMI

다음 copy-image 명령은 --copy-image-tags 파라미터를 사용하여 를 복사할 때 사용자 정의 AMI 태그를 복사합니다AMI.

```
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
```

출력:

```
{
    "ImageId": "ami-0123456789abcdefg"
}
```

자세한 내용은 Amazon EC2 사용 설명서의 복사AMI를 참조하세요.

- 자세한 API 내용은 명령 참조CopyImage의 섹션을 참조하세요. AWS CLI

### copy-snapshot

다음 코드 예시에서는 copy-snapshot을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 스냅샷을 다른 리전에 복사하는 방법

다음 copy-snapshot 예제 명령은 지정된 스냅샷을 us-west-2 리전에서 us-east-1 리전으로 복사하고 간단한 설명을 추가합니다.

```
aws ec2 copy-snapshot \
    --region us-east-1 \
    --source-region us-west-2 \
    --source-snapshot-id snap-066877671789bd71b \
    --description "This is my copied snapshot."
```

출력:

```
{
    "SnapshotId": "snap-066877671789bd71b"
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon EBS 스냅샷 복사를 참조하세요. EC2

예제 2: 암호화되지 않은 스냅샷을 복사하고 새 스냅샷을 암호화하는 방법

다음 copy-snapshot 명령은 지정된 암호화되지 않은 스냅샷을 us-west-2 리전에서 현재 리전으로 복사하고 지정된 KMS 키를 사용하여 새 스냅샷을 암호화합니다.

```
aws ec2 copy-snapshot \
    --source-region us-west-2 \
    --source-snapshot-id snap-066877671789bd71b \
    --encrypted \
    --kms-key-id alias/my-kms-key
```

출력:

```
{
    "SnapshotId": "snap-066877671789bd71b"
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon EBS 스냅샷 복사를 참조하세요. EC2

- 자세한 API 내용은 명령 참조CopySnapshot의 섹션을 참조하세요. AWS CLI

### create-capacity-reservation-fleet

다음 코드 예시에서는 create-capacity-reservation-fleet을 사용하는 방법을 보여 줍니다.

AWS CLI

용량 예약 플릿을 생성하려면

다음 create-capacity-reservation-fleet 예제에서는 요청에 지정된 인스턴스 유형에 대해 지정된 총 목표 용량까지 용량 예약 플릿을 생성합니다. 용량 예약 플릿이 용량을 예약하는 인스턴스 수는 요청에 지정하는 총 목표 용량 및 인스턴스 유형 가중치에 따라 달라집니다. 사용할 인스턴스 유형과 지정된 각 인스턴스 유형에 대한 우선 순위를 지정합니다.

```
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
```

instanceTypeSpecification.json의 콘텐츠:

```
[
    {
        "InstanceType": "m5.xlarge",
        "InstancePlatform": "Linux/UNIX",
        "Weight": 3.0,
        "AvailabilityZone":"us-east-1a",
        "EbsOptimized": true,
        "Priority" : 1
    }
]
```

출력:

```
{
    "Status": "submitted",
    "TotalFulfilledCapacity": 0.0,
    "CapacityReservationFleetId": "crf-abcdef01234567890",
    "TotalTargetCapacity": 24
}
```

용량 예약 플릿에 대한 자세한 내용은 Amazon EC2 사용 설명서의 용량 예약 플릿을 참조하세요.

인스턴스 유형 가중치 및 총 목표 용량에 대한 자세한 내용은 Amazon EC2 사용 설명서의 인스턴스 유형 가중치 및 총 목표 용량을 참조하세요.

지정된 인스턴스 유형에 대한 우선 순위 지정에 대한 자세한 내용은 Amazon EC2 사용 설명서의 할당 전략 및 인스턴스 유형 우선 순위를 참조하세요.

- 자세한 API 내용은 명령 참조CreateCapacityReservationFleet의 섹션을 참조하세요. AWS CLI

### `create-capacity-reservation`

다음 코드 예시에서는 `create-capacity-reservation`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 용량 예약 생성

다음 `create-capacity-reservation` 예제에서는 `eu-west-1a` 가용 영역에 용량 예약을 생성하여 Linux/Unix 운영 체제를 실행하는 `t2.medium` 인스턴스 3개를 시작할 수 있습니다. 기본적으로 용량 예약은 열린 인스턴스 일치 기준으로 생성되며 임시 스토리지를 지원하지 않으며 수동으로 취소할 때까지 활성 상태로 유지됩니다.

```
aws ec2 create-capacity-reservation \
    --availability-zone eu-west-1a \
    --instance-type t2.medium \
    --instance-platform Linux/UNIX \
    --instance-count 3
```

출력:

```
{
    "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"
    }
}
```

예제 2: 지정된 날짜/시간에 자동으로 종료되는 용량 예약을 생성하려면

다음 create-capacity-reservation 예제에서는 eu-west-1a 가용 영역에 용량 예약을 생성하여 Linux/Unix 운영 체제를 실행하는 m5.large 인스턴스 3개를 시작할 수 있습니다. 이 용량 예약은 08/31/2019 23:59:59에 자동으로 종료됩니다.

```
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
```

출력:

```
{
    "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"
    }
}
```

예제 3: 대상 인스턴스 시작만 허용하는 용량 예약을 생성하려면

다음 create-capacity-reservation 예제에서는 대상 인스턴스 시작만 허용하는 용량 예약을 생성합니다.

```
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
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 용량 예약 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateCapacityReservation의 섹션을 참조하세요. AWS CLI

### create-carrier-gateway

다음 코드 예시에서는 create-carrier-gateway을 사용하는 방법을 보여 줍니다.

AWS CLI

캐리어 게이트웨이를 생성하려면

다음 create-carrier-gateway 예제에서는 지정된 에 대한 캐리어 게이트웨이를 생성합니다 VPC.

```
aws ec2 create-carrier-gateway \
    --vpc-id vpc-0c529aEXAMPLE1111
```

출력:

```
{
    "CarrierGateway": {
        "CarrierGatewayId": "cagw-0465cdEXAMPLE1111",
        "VpcId": "vpc-0c529aEXAMPLE1111",
        "State": "pending",
        "OwnerId": "123456789012"
    }
}
```

자세한 내용은 AWS Wavelength 사용 설명서의 캐리어 게이트웨이를 참조하세요.

- 자세한 API 내용은 명령 참조CreateCarrierGateway의 섹션을 참조하세요. AWS CLI

### `create-client-vpn-endpoint`

다음 코드 예시에서는 `create-client-vpn-endpoint`을 사용하는 방법을 보여 줍니다.

AWS CLI

클라이언트 VPN 엔드포인트를 생성하려면

다음 `create-client-vpn-endpoint` 예제에서는 상호 인증을 사용하고 클라이언트 CIDR 블록의 값을 지정하는 클라이언트 VPN 엔드포인트를 생성합니다.

```
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
```

출력:

```
{
    "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",
    "Status": {
        "Code": "pending-associate"
    },
```

```
    "DnsName": "cvpn-endpoint-123456789123abcde.prod.clientvpn.ap-
south-1.amazonaws.com"
}
```

자세한 내용은 클라이언트 관리자 안내서의 클라이언트 VPN 엔드포인트를 참조하세요. AWS VPN

- 자세한 API 내용은 명령 참조CreateClientVpnEndpoint의 섹션을 참조하세요. AWS CLI

### `create-client-vpn-route`

다음 코드 예시에서는 `create-client-vpn-route`을 사용하는 방법을 보여 줍니다.

AWS CLI

클라이언트 VPN 엔드포인트에 대한 경로를 생성하려면

다음 `create-client-vpn-route` 예제에서는 클라이언트 VPN 엔드포인트의 지정된 서브넷에 대한 경로를 인터넷(`0.0.0.0/0`)에 추가합니다.

```
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
```

출력:

```
{
    "Status": {
        "Code": "creating"
    }
}
```

자세한 내용은 AWS 클라이언트 VPN 관리자 안내서의 라우팅을 참조하세요.

- 자세한 API 내용은 명령 참조CreateClientVpnRoute의 섹션을 참조하세요. AWS CLI

### `create-coip-cidr`

다음 코드 예시에서는 `create-coip-cidr`을 사용하는 방법을 보여 줍니다.

AWS CLI

고객 소유 IP(CoIP) 주소 범위를 생성하려면

다음 create-coip-cidr 예제에서는 지정된 CoIP 풀에 지정된 범위의 CoIP 주소를 생성합니다.

```
aws ec2 create-coip-cidr \
    --cidr 15.0.0.0/24 \
    --coip-pool-id ipv4pool-coip-1234567890abcdefg
```

출력:

```
{
    "CoipCidr": {
        "Cidr": "15.0.0.0/24",
        "CoipPoolId": "ipv4pool-coip-1234567890abcdefg",
        "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890"
    }
}
```

자세한 내용은 AWS Outposts 사용 설명서의 고객 소유 IP 주소를 참조하세요.

- 자세한 API 내용은 명령 참조CreateCoipCidr의 섹션을 참조하세요. AWS CLI

### create-coip-pool

다음 코드 예시에서는 create-coip-pool을 사용하는 방법을 보여 줍니다.

AWS CLI

고객 소유 IP(CoIP) 주소 풀을 생성하려면

다음 create-coip-pool 예제에서는 지정된 로컬 게이트웨이 라우팅 테이블에 CoIP 주소에 대한 CoIP 풀을 생성합니다.

```
aws ec2 create-coip-pool \
    --local-gateway-route-table-id lgw-rtb-abcdefg1234567890
```

출력:

```
{
    "CoipPool": {
        "PoolId": "ipv4pool-coip-1234567890abcdefg",
        "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890",
        "PoolArn": "arn:aws:ec2:us-west-2:123456789012:coip-pool/ipv4pool-
coip-1234567890abcdefg"
```

```
    }
}
```

자세한 내용은 AWS Outposts 사용 설명서의 고객 소유 IP 주소를 참조하세요.

- 자세한 API 내용은 명령 참조CreateCoipPool의 섹션을 참조하세요. AWS CLI

### `create-customer-gateway`

다음 코드 예시에서는 `create-customer-gateway`을 사용하는 방법을 보여 줍니다.

AWS CLI

고객 게이트웨이를 생성하려면

이 예제에서는 외부 인터페이스에 지정된 IP 주소를 사용하여 고객 게이트웨이를 생성합니다.

명령:

```
aws ec2 create-customer-gateway --type ipsec.1 --public-ip 12.1.2.3 --bgp-asn 65534
```

출력:

```
{
    "CustomerGateway": {
        "CustomerGatewayId": "cgw-0e11f167",
        "IpAddress": "12.1.2.3",
        "State": "available",
        "Type": "ipsec.1",
        "BgpAsn": "65534"
    }
}
```

- 자세한 API 내용은 명령 참조CreateCustomerGateway의 섹션을 참조하세요. AWS CLI

### `create-default-subnet`

다음 코드 예시에서는 `create-default-subnet`을 사용하는 방법을 보여 줍니다.

AWS CLI

기본 서브넷을 생성하려면

이 예제에서는 가용 영역 에 기본 서브넷을 생성합니다us-east-2a.

명령:

```
 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
   }
 }
```

- 자세한 API 내용은 명령 참조CreateDefaultSubnet의 섹션을 참조하세요. AWS CLI

### `create-default-vpc`

다음 코드 예시에서는 `create-default-vpc`을 사용하는 방법을 보여 줍니다.

AWS CLI

기본값을 생성하려면 VPC

이 예제에서는 기본 을 생성합니다VPC.

명령:

```
aws ec2 create-default-vpc
```

출력:

```
{
```

```
    "Vpc": {
        "VpcId": "vpc-8eaae5ea",
        "InstanceTenancy": "default",
        "Tags": [],
        "Ipv6CidrBlockAssociationSet": [],
        "State": "pending",
        "DhcpOptionsId": "dopt-af0c32c6",
        "CidrBlock": "172.31.0.0/16",
        "IsDefault": true
    }
}
```

- 자세한 API 내용은 명령 참조CreateDefaultVpc의 섹션을 참조하세요. AWS CLI

### `create-dhcp-options`

다음 코드 예시에서는 `create-dhcp-options`을 사용하는 방법을 보여 줍니다.

AWS CLI

DHCP 옵션 집합을 생성하려면

다음 `create-dhcp-options` 예제에서는 도메인 이름, 도메인 이름 서버 및 NetBIOS 노드 유형을 지정하는 DHCP 옵션 세트를 생성합니다.

```
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"
```

출력:

```
{
    "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"
    }
}
```

- 자세한 API 내용은 명령 참조CreateDhcpOptions의 섹션을 참조하세요. AWS CLI

### `create-egress-only-internet-gateway`

다음 코드 예시에서는 `create-egress-only-internet-gateway`을 사용하는 방법을 보여 줍니다.

AWS CLI

송신 전용 인터넷 게이트웨이를 생성하려면

이 예제에서는 지정된 에 대한 송신 전용 인터넷 게이트웨이를 생성합니다VPC.

명령:

```
aws ec2 create-egress-only-internet-gateway --vpc-id vpc-0c62a468
```

출력:

```
{
  "EgressOnlyInternetGateway": {
      "EgressOnlyInternetGatewayId": "eigw-015e0e244e24dfe8a",
      "Attachments": [
          {
              "State": "attached",
              "VpcId": "vpc-0c62a468"
          }
      ]
  }
}
```

- 자세한 API 내용은 명령 참조CreateEgressOnlyInternetGateway의 섹션을 참조하세요. AWS CLI

### `create-fleet`

다음 코드 예시에서는 `create-fleet`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 스팟 인스턴스를 기본 구매 모델로 시작하는 EC2 플릿 생성

다음 `create-fleet` 예제에서는 EC2플릿을 시작하는 데 필요한 최소 파라미터인 시작 템플릿, 대상 용량 및 기본 구매 모델을 사용하여 플릿을 생성합니다. 시작 템플릿은 시작 템플릿 ID와 버전 번호로 식별됩니다. 플릿의 목표 용량은 인스턴스 2개이고 기본 구매 모델은 로spot, 플릿이 스팟 인스턴스 2개를 시작합니다.

EC2 플릿을 생성할 때 JSON 파일을 사용하여 시작할 인스턴스에 대한 정보를 지정합니다.

```
aws ec2 create-fleet \
    --cli-input-json file://file_name.json
```

file_name.json의 내용:

```
{
    "LaunchTemplateConfigs": [
    {
        "LaunchTemplateSpecification": {
        "LaunchTemplateId": "lt-0e8c754449b27161c",
```

```
        "Version": "1"
        }
    }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 2,
        "DefaultTargetCapacityType": "spot"
    }
}
```

출력:

```
{
    "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
}
```

예제 2: 온디맨드 인스턴스를 기본 구매 모델로 시작하는 EC2 플릿을 생성하는 방법

다음 `create-fleet` 예제에서는 EC2플릿을 시작하는 데 필요한 최소 파라미터인 시작 템플릿, 대상 용량 및 기본 구매 모델을 사용하여 플릿을 생성합니다. 시작 템플릿은 시작 템플릿 ID와 버전 번호로 식별됩니다. 플릿의 목표 용량은 인스턴스 2개이고 기본 구매 모델은 로on-demand, 플릿이 온디맨드 인스턴스 2개를 시작합니다.

EC2 플릿을 생성할 때 JSON 파일을 사용하여 시작할 인스턴스에 대한 정보를 지정합니다.

```
aws ec2 create-fleet \
    --cli-input-json file://file_name.json
```

file_name.json의 내용:

```
{
    "LaunchTemplateConfigs": [
    {
        "LaunchTemplateSpecification": {
        "LaunchTemplateId": "lt-0e8c754449b27161c",
        "Version": "1"
        }
    }
    ],
    "TargetCapacitySpecification": {
    "TotalTargetCapacity": 2,
    "DefaultTargetCapacityType": "on-demand"
```

```
    }
}
```

출력:

```
{
    "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
}
```

예제 3: 온디맨드 인스턴스를 기본 용량으로 시작하는 EC2 플릿을 생성하려면

다음 `create-fleet` 예제에서는 EC2 플릿의 총 목표 용량인 인스턴스 2개와 온디맨드 인스턴스 1개의 목표 용량을 지정하는 플릿을 생성합니다. 기본 구매 모델은 입니다`spot`. 플릿은 지정된 대로 온디맨드 인스턴스 1개를 시작하지만 총 목표 용량을 충족하려면 인스턴스를 하나 더 시작해야 합니다. 차이에 대한 구매 모델은 `TotalTargetCapacity - OnDemandTargetCapacity = `로 계산`DefaultTargetCapacityType`되어 플릿이 1 스팟 인스턴스를 시작합니다.

EC2 플릿을 생성할 때 JSON 파일을 사용하여 시작할 인스턴스에 대한 정보를 지정합니다.

```
aws ec2 create-fleet \
    --cli-input-json file://file_name.json
```

file_name.json의 내용:

```
{
    "LaunchTemplateConfigs": [
    {
        "LaunchTemplateSpecification": {
        "LaunchTemplateId": "lt-0e8c754449b27161c",
        "Version": "1"
        }
    }
    ],
    "TargetCapacitySpecification": {
        "TotalTargetCapacity": 2,
        "OnDemandTargetCapacity":1,
        "DefaultTargetCapacityType": "spot"
    }
}
```

출력:

```
{
    "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
}
```

예제 4: 최저 가격 할당 전략을 사용하여 스팟 인스턴스를 시작하는 EC2 플릿을 생성하려면

스팟 인스턴스의 할당 전략이 지정되어 있지 않으면 기본 할당 전략인 `lowest-price`가 사용됩니다. 다음 `create-fleet` 예제에서는 `lowest-price` 할당 전략을 사용하여 EC2 플릿을 생성합니다. 시작 템플릿을 재정의하고 서로 인스턴스 유형은 다르지만 가중치 용량과 서브넷이 동일한 시작 사양 3개가 있습니다. 총 목표 용량은 인스턴스 2개이고 기본 구매 모델은 입니다`spot`. EC2 플릿은 최저 가격으로 시작 사양의 인스턴스 유형을 사용하여 2개의 스팟 인스턴스를 시작합니다.

EC2 플릿을 생성할 때 JSON 파일을 사용하여 시작할 인스턴스에 대한 정보를 지정합니다.

```
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"

  }
}
```

출력:

```
{
    "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
}
```

- 자세한 API 내용은 명령 참조CreateFleet의 섹션을 참조하세요. AWS CLI

### `create-flow-logs`

다음 코드 예시에서는 `create-flow-logs`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 흐름 로그 생성

다음 `create-flow-logs` 예제에서는 지정된 네트워크 인터페이스에 대해 거부된 모든 트래픽을 캡처하는 흐름 로그를 생성합니다. 흐름 로그는 지정된 IAM 역할의 권한을 사용하여 CloudWatch 로그의 로그 그룹에 전달됩니다.

```
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
```

출력:

```
{
    "ClientToken": "so0eNA2uSHUNlHI0S2cJ305GuIX1CezaRdGtexample",
    "FlowLogIds": [
        "fl-12345678901234567"
```

```
    ],
    "Unsuccessful": []
}
```

자세한 내용은 Amazon VPC 사용 설명서의 VPC 흐름 로그를 참조하세요.

예제 2: 사용자 지정 형식으로 흐름 로그 생성

다음 `create-flow-logs` 예제에서는 지정된 의 모든 트래픽을 캡처VPC하고 Amazon S3 버킷에 흐름 로그를 전달하는 흐름 로그를 생성합니다. `--log-format` 파라미터는 흐름 로그 레코드의 사용자 지정 형식을 지정합니다. Windows에서 이 명령을 실행하려면 작은따옴표(')를 큰따옴표(")로 변경합니다.

```
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}'
```

자세한 내용은 Amazon VPC 사용 설명서의 VPC 흐름 로그를 참조하세요.

예제 3: 최대 집계 간격이 1분인 흐름 로그 생성

다음 `create-flow-logs` 예제에서는 지정된 의 모든 트래픽을 캡처VPC하고 Amazon S3 버킷에 흐름 로그를 전달하는 흐름 로그를 생성합니다. `--max-aggregation-interval` 파라미터는 60초(1분)의 최대 집계 간격을 지정합니다.

```
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
```

자세한 내용은 Amazon VPC 사용 설명서의 VPC 흐름 로그를 참조하세요.

- 자세한 API 내용은 명령 참조CreateFlowLogs의 섹션을 참조하세요. AWS CLI

### create-fpga-image

다음 코드 예시에서는 create-fpga-image을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon FPGA 이미지를 생성하려면

이 예제에서는 지정된 버킷의 지정된 tarballAFI에서 를 생성합니다.

명령:

```
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
```

출력:

```
{
  "FpgaImageId": "afi-0d123e123bfc85abc",
  "FpgaImageGlobalId": "agfi-123cb27b5e84a0abc"
}
```

- 자세한 API 내용은 명령 참조CreateFpgaImage의 섹션을 참조하세요. AWS CLI

### create-image

다음 코드 예시에서는 create-image을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: Amazon EBS지원 인스턴스AMI에서 를 생성하려면

다음 create-image 예제에서는 지정된 인스턴스AMI에서 를 생성합니다.

```
aws ec2 create-image \
    --instance-id i-1234567890abcdef0 \
    --name "My server" \
    --description "An AMI for my server"
```

출력:

```
{
    "ImageId": "ami-abcdef01234567890"
}
```

에 대한 블록 디바이스 매핑을 지정하는 방법에 대한 자세한 내용은 Amazon EC2 사용 설명서의 에 대한 블록 디바이스 매핑 지정AMI을 AMI참조하세요.

예제 2: 재부팅 없이 Amazon EBS지원 인스턴스AMI에서 를 생성하려면

다음 `create-image` 예제에서는 이미지를 생성하기 전에 인스턴스가 재부팅되지 않도록 를 생성하고 --no-reboot 파라미터를 AMI 설정합니다.

```
aws ec2 create-image \
    --instance-id i-1234567890abcdef0 \
    --name "My server" \
    --no-reboot
```

출력:

```
{
    "ImageId": "ami-abcdef01234567890"
}
```

에 대한 블록 디바이스 매핑을 지정하는 방법에 대한 자세한 내용은 Amazon EC2 사용 설명서의 에 대한 블록 디바이스 매핑 지정AMI을 AMI참조하세요.

예제 3: 생성 시 AMI 및 스냅샷에 태그 지정

다음 `create-image` 예제에서는 를 생성하고 AMI및 스냅샷AMI에 동일한 태그를 지정합니다. `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}]"
```

출력:

```
{
    "ImageId": "ami-abcdef01234567890"
```

```
}
```

생성 시 리소스 태그 지정에 대한 자세한 내용은 Amazon EC2 사용 설명서의 리소스 생성에 태그 추가를 참조하세요.

- 자세한 API 내용은 명령 참조CreateImage의 섹션을 참조하세요. AWS CLI

### `create-instance-connect-endpoint`

다음 코드 예시에서는 `create-instance-connect-endpoint`을 사용하는 방법을 보여 줍니다.

AWS CLI

EC2 인스턴스 연결 엔드포인트를 생성하려면

다음 `create-instance-connect-endpoint` 예제에서는 지정된 서브넷에 EC2 인스턴스 연결 엔드포인트를 생성합니다.

```
aws ec2 create-instance-connect-endpoint \
    --region us-east-1 \
    --subnet-id subnet-0123456789example
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon EC2 사용 설명서의 EC2 인스턴스 연결 엔드포인트 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateInstanceConnectEndpoint의 섹션을 참조하세요. AWS CLI

### `create-instance-event-window`

다음 코드 예시에서는 `create-instance-event-window`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 시간 범위를 가진 이벤트 창을 생성하려면

다음 `create-instance-event-window` 예제에서는 시간 범위가 있는 이벤트 창을 생성합니다. `cron-expression` 파라미터를 함께 지정할 수는 없습니다.

```
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
```

출력:

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "TimeRanges": [
            {
                "StartWeekDay": "monday",
                "StartHour": 2,
                "EndWeekDay": "wednesday",
                "EndHour": 8
            }
        ],
```

```
        "Name": "myEventWindowName",
        "State": "creating",
        "Tags": [
            {
                "Key": "K1",
                "Value": "V1"
            }
        ]
    }
}
```

이벤트 기간 제약 조건은 Amazon EC2 사용 설명서의 예약된 이벤트 섹션의 고려 사항을 참조하세요.

예제 2: cron 표현식을 사용하여 이벤트 창을 생성하려면

다음 `create-instance-event-window` 예제에서는 cron 표현식을 사용하여 이벤트 창을 생성합니다. `time-range` 파라미터를 함께 지정할 수는 없습니다.

```
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
```

출력:

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "State": "creating",
        "Tags": [
            {
                "Key": "K1",
                "Value": "V1"
            }
        ]
    }
}
```

이벤트 기간 제약 조건은 Amazon EC2 사용 설명서의 예약된 이벤트 섹션의 고려 사항을 참조하세요.

- 자세한 API 내용은 명령 참조CreateInstanceEventWindow의 섹션을 참조하세요. AWS CLI

### `create-instance-export-task`

다음 코드 예시에서는 `create-instance-export-task`을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스를 내보내려면

이 예제 명령은 인스턴스 i-1234567890abcdef0을 Amazon S3 버킷 myexportbucket으로 내보내는 작업을 생성합니다.

명령:

```
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
```

출력:

```
{
    "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"
    }
}
```

- 자세한 API 내용은 명령 참조CreateInstanceExportTask의 섹션을 참조하세요. AWS CLI

### create-internet-gateway

다음 코드 예시에서는 create-internet-gateway을 사용하는 방법을 보여 줍니다.

AWS CLI

인터넷 게이트웨이를 생성하려면

다음 create-internet-gateway 예제에서는 태그 를 사용하여 인터넷 게이트웨이를 생성합니다Name=my-igw.

```
aws ec2 create-internet-gateway \
    --tag-specifications ResourceType=internet-gateway,Tags=[{Key=Name,Value=my-igw}]
```

출력:

```
{
    "InternetGateway": {
        "Attachments": [],
        "InternetGatewayId": "igw-0d0fb496b3994d755",
        "OwnerId": "123456789012",
        "Tags": [
            {
                "Key": "Name",
                "Value": "my-igw"
            }
        ]
    }
}
```

자세한 내용은 Amazon VPC 사용 설명서의 인터넷 게이트웨이를 참조하세요.

- 자세한 API 내용은 명령 참조CreateInternetGateway의 섹션을 참조하세요. AWS CLI

### create-ipam-pool

다음 코드 예시에서는 create-ipam-pool을 사용하는 방법을 보여 줍니다.

AWS CLI

IPAM 풀을 생성하려면

다음 `create-ipam-pool` 예제에서는 IPAM 풀을 생성합니다.

(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"}]
```

출력:

```
{
    "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"
            }
        ]
    }
}
```

자세한 내용은 Amazon VPC IPAM 사용 설명서의 IP 주소 프로비저닝 계획을 참조하세요.

- 자세한 API 내용은 명령 참조CreateIpamPool의 섹션을 참조하세요. AWS CLI

### `create-ipam-resource-discovery`

다음 코드 예시에서는 `create-ipam-resource-discovery`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 검색을 생성하려면

이 예제에서는 다른 AWS 조직의 IPAM 관리자가 조직 IPAM 내 리소스의 IP 주소를 관리하고 모니터링할 수 있도록 리소스 검색을 생성하고 다른 조직의 관리자와 공유하려는 위임된 관리자입니다.

중요

이 예제에는 `--region` 및 `--operating-regions` 옵션이 모두 포함되어 있습니다. 리소스 검색을 와 성공적으로 통합하려면 옵션도 선택 사항이지만 특정 방식으로 구성해야 하기 때문입니다 IPAM. * 는 검색IPAM하려는 리소스가 있는 리전과 일치해야 `--operating-regions` 합니다. IP 주소를 관리하지 않으IPAM려는 리전이 있는 경우(예: 규정 준수 이유로 인해) 포함하지 마세요. *는 IP 주소를 연결IPAM하려는 의 홈 리전과 일치해야 `--region` 합니다. 가 생성된 동일한 리전에서

리소스 검색을 생성IPAM해야 합니다. 예를 들어, 연결IPAM하려는 이 us-east-1에서 생성된 경우 요청에 `--region us-east-1` 를 포함합니다. `--region` 및 `--operating-regions` 옵션은 모두 지정하지 않으면 명령을 실행 중인 리전으로 기본 설정됩니다.

이 예제에서는 통합 IPAM 중인 의 운영 리전에 `us-west-1`, 및 `us-west-2`가 포함됩니다`ap-south-1`. 리소스 검색을 생성할 때 `us-west-1` 및 에서 리소스 IP 주소를 검색IPAM하고 에서는 검색`us-west-2`하지 않으려고 합니다`ap-south-1`. 따라서 `--operating-regions RegionName='us-west-1' RegionName='us-west-2'` 요청에만 포함됩니다.

다음 `create-ipam-resource-discovery` 예제에서는 IPAM 리소스 검색을 생성합니다.

```
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
```

출력:

```
{
    "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"
            }
        ]
}
```

리소스 검색을 생성한 후에는 로 수행할 수 있는 다른 IPAM 위임된 관리자와 공유할 수 있습니다create-resource-share. 자세한 내용은 Amazon VPC IPAM 사용 설명서의 조직 외부 계정IPAM과 통합을 참조하세요.

- 자세한 API 내용은 명령 참조CreateIpamResourceDiscovery의 섹션을 참조하세요. AWS CLI

### `create-ipam-scope`

다음 코드 예시에서는 `create-ipam-scope`을 사용하는 방법을 보여 줍니다.

AWS CLI

IPAM 범위를 생성하려면

다음 `create-ipam-scope` 예제에서는 IPAM 범위를 생성합니다.

(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"}]
```

출력:

```
{
    "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"
            }
        ]
    }
}
```

자세한 내용은 Amazon VPC IPAM 사용 설명서의 추가 범위 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateIpamScope의 섹션을 참조하세요. AWS CLI

### `create-ipam`

다음 코드 예시에서는 `create-ipam`을 사용하는 방법을 보여 줍니다.

AWS CLI

를 생성하려면 IPAM

다음 `create-ipam` 예제에서는 를 생성합니다IPAM.

(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}]
```

출력:

```
{
    "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"
            }
        ]
    }
}
```

자세한 내용은 Amazon VPC IPAM 사용 설명서의 생성 IPAM 섹션을 참조하세요.

- 자세한 API 내용은 명령 참조CreateIpam의 섹션을 참조하세요. AWS CLI

### `create-key-pair`

다음 코드 예시에서는 `create-key-pair`을 사용하는 방법을 보여 줍니다.

AWS CLI

키 페어를 생성하는 방법

이 예제에서는 이름이 MyKeyPair인 키 페어를 생성합니다.

명령:

```
aws ec2 create-key-pair --key-name MyKeyPair
```

출력은 프라이빗 키 및 키 지문의 ASCII 버전입니다. 키는 파일에 저장해야 합니다.

자세한 내용은 AWS Command Line Interface 사용 설명서의 키 페어 사용을 참조하세요.

- 자세한 API 내용은 명령 참조CreateKeyPair의 섹션을 참조하세요. AWS CLI

### create-launch-template-version

다음 코드 예시에서는 create-launch-template-version을 사용하는 방법을 보여 줍니다.

AWS CLI

시작 템플릿 버전을 생성하려면

이 예제에서는 시작 템플릿의 버전 1을 기반으로 새 시작 템플릿 버전을 생성하고 다른 AMI ID를 지정합니다.

명령:

```
aws ec2 create-launch-template-version --launch-template-id lt-0abcd290751193123
 --version-description WebVersion2 --source-version 1 --launch-template-data
 '{"ImageId":"ami-c998b6b2"}'
```

출력:

```
{
  "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"
    }
}
```

- 자세한 API 내용은 명령 참조CreateLaunchTemplateVersion의 섹션을 참조하세요. AWS CLI

### create-launch-template

다음 코드 예시에서는 create-launch-template을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: 시작 템플릿을 생성하는 방법

다음 create-launch-template 예제에서는 인스턴스를 시작할 서브넷을 지정하는 시작 템플릿을 생성하고, 인스턴스에 퍼블릭 IP 주소와 IPv6 주소를 할당하고, 인스턴스에 대한 태그를 생성합니다.

```
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"}]}]}'
```

출력:

```
{
```

```
    "LaunchTemplate": {
        "LatestVersionNumber": 1,
        "LaunchTemplateId": "lt-01238c059e3466abc",
        "LaunchTemplateName": "TemplateForWebServer",
        "DefaultVersionNumber": 1,
        "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
        "CreateTime": "2019-01-27T09:13:24.000Z"
    }
}
```

자세한 내용은 Amazon Elastic Compute Cloud 사용 설명서의 시작 템플릿에서 인스턴스 시작을 참조하세요. JSON형식이 지정된 파라미터의 인용에 대한 자세한 내용은 AWS 명령줄 인터페이스 사용 설명서의 문자열 인용을 참조하세요.

예제 2: Amazon EC2 Auto Scaling에 대한 시작 템플릿 생성

다음 `create-launch-template` 예제에서는 여러 태그가 있는 시작 템플릿과 블록 디바이스 매핑을 생성하여 인스턴스가 시작될 때 추가 EBS 볼륨을 지정합니다. VPC Auto Scaling 그룹이 인스턴스를 시작할 의 보안 그룹에 `Groups` 해당하는 값을 지정합니다. VPC 및 서브넷을 Auto Scaling 그룹의 속성으로 지정합니다.

```
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
```

출력:

```
{
    "LaunchTemplate": {
        "LatestVersionNumber": 1,
        "LaunchTemplateId": "lt-0123c79c33a54e0abc",
        "LaunchTemplateName": "TemplateForAutoScaling",
        "DefaultVersionNumber": 1,
```

```
        "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
        "CreateTime": "2019-04-30T18:16:06.000Z"
    }
}
```

자세한 내용은 Amazon Auto Scaling 사용 설명서의 Auto Scaling 그룹에 대한 시작 템플릿 생성을 참조하세요.  EC2 Auto Scaling  JSON형식이 지정된 파라미터의 인용에 대한 자세한 내용은 AWS 명령줄 인터페이스 사용 설명서의 문자열 인용을 참조하세요.

예제 3: EBS 볼륨 암호화를 지정하는 시작 템플릿을 생성하려면

다음 `create-launch-template` 예제에서는 암호화되지 않은 스냅샷에서 생성된 암호화된 EBS 볼륨을 포함하는 시작 템플릿을 생성합니다. 또한 생성 중에 볼륨에 태그도 지정합니다. 기본적으로 암호화가 비활성화된 경우 다음 예제에 표시된 대로 `"Encrypted"` 옵션을 지정해야 합니다. `"KmsKeyId"` 옵션을 사용하여 고객 관리형 를 지정하는 경우 기본적으로 암호화가 활성화되어 있더라도 `"Encrypted"` 옵션을 지정CMK해야 합니다.

```
aws ec2 create-launch-template \
  --launch-template-name TemplateForEncryption \
  --launch-template-data file://config.json
```

config.json의 콘텐츠:

```
{
    "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"
                }
            ]
        }
    ]
}
```

출력:

```
{
    "LaunchTemplate": {
        "LatestVersionNumber": 1,
        "LaunchTemplateId": "lt-0d5bd51bcf8530abc",
        "LaunchTemplateName": "TemplateForEncryption",
        "DefaultVersionNumber": 1,
        "CreatedBy": "arn:aws:iam::123456789012:user/Bob",
        "CreateTime": "2020-01-07T19:08:36.000Z"
    }
}
```

자세한 내용은 Amazon Elastic Compute Cloud 사용 설명서의 스냅샷에서 Amazon EBS 볼륨 복원 및 기본값으로 암호화를 참조하세요.

- 자세한 API 내용은 명령 참조CreateLaunchTemplate의 섹션을 참조하세요. AWS CLI

### `create-local-gateway-route-table-virtual-interface-group-association`

다음 코드 예시에서는 `create-local-gateway-route-table-virtual-interface-group-association`을 사용하는 방법을 보여 줍니다.

AWS CLI

로컬 게이트웨이 라우팅 테이블을 가상 인터페이스(VIFs) 그룹에 연결하려면

다음 `create-local-gateway-route-table-virtual-interface-group-association` 예제에서는 지정된 로컬 게이트웨이 라우팅 테이블과 VIF 그룹 간의 연결을 생성합니다.

```
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
```

출력:

```
{
    "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": []
    }
}
```

자세한 내용은 Outposts 사용 설명서의 VIF 그룹 연결을 참조하세요. AWS

- 자세한 API 내용은 명령 참조CreateLocalGatewayRouteTableVirtualInterfaceGroupAssociation의 섹션을 참조하세요. AWS CLI

### create-local-gateway-route-table-vpc-association

다음 코드 예시에서는 create-local-gateway-route-table-vpc-association을 사용하는 방법을 보여 줍니다.

AWS CLI

VPC를 라우팅 테이블에 연결하려면

다음 create-local-gateway-route-table-vpc-association 예제에서는 지정된 를 지정된 로컬 게이트웨이 라우팅 테이블VPC과 연결합니다.

```
aws ec2 create-local-gateway-route-table-vpc-association \
    --local-gateway-route-table-id lgw-rtb-059615ef7dEXAMPLE \
    --vpc-id vpc-07ef66ac71EXAMPLE
```

출력:

```
{
    "LocalGatewayRouteTableVpcAssociation": {
        "LocalGatewayRouteTableVpcAssociationId": "lgw-vpc-assoc-0ee765bcc8EXAMPLE",
        "LocalGatewayRouteTableId": "lgw-rtb-059615ef7dEXAMPLE",
        "LocalGatewayId": "lgw-09b493aa7cEXAMPLE",
        "VpcId": "vpc-07ef66ac71EXAMPLE",
        "State": "associated"
    }
}
```

- 자세한 API 내용은 명령 참조CreateLocalGatewayRouteTableVpcAssociation의 섹션을 참조하세요. AWS CLI

### `create-local-gateway-route-table`

다음 코드 예시에서는 `create-local-gateway-route-table`을 사용하는 방법을 보여 줍니다.

AWS CLI

로컬 게이트웨이 라우팅 테이블을 생성하려면

다음 `create-local-gateway-route-table` 예제에서는 직접 라우팅 모드를 사용하여 로컬 게이트웨이 VPC 라우팅 테이블을 생성합니다.

```
aws ec2 create-local-gateway-route-table \
    --local-gateway-id lgw-1a2b3c4d5e6f7g8h9 \
    --mode direct-vpc-routing
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS Outposts 사용 설명서의 로컬 게이트웨이 라우팅 테이블을 참조하세요.

- 자세한 API 내용은 명령 참조CreateLocalGatewayRouteTable의 섹션을 참조하세요. AWS CLI

### `create-local-gateway-route`

다음 코드 예시에서는 `create-local-gateway-route`을 사용하는 방법을 보여 줍니다.

AWS CLI

로컬 게이트웨이 라우팅 테이블에 대한 정적 라우팅을 생성하려면

다음 `create-local-gateway-route` 예제에서는 지정된 로컬 게이트웨이 라우팅 테이블에 지정된 라우팅을 생성합니다.

```
aws ec2 create-local-gateway-route \
    --destination-cidr-block 0.0.0.0/0 \
    --local-gateway-route-table-id lgw-rtb-059615ef7dEXAMPLE
```

출력:

```
{
    "Route": {
        "DestinationCidrBlock": "0.0.0.0/0",
        "LocalGatewayVirtualInterfaceGroupId": "lgw-vif-grp-07145b276bEXAMPLE",
        "Type": "static",
        "State": "deleted",
        "LocalGatewayRouteTableId": "lgw-rtb-059615ef7dEXAMPLE"
    }
}
```

- 자세한 API 내용은 명령 참조CreateLocalGatewayRoute의 섹션을 참조하세요. AWS CLI

### `create-managed-prefix-list`

다음 코드 예시에서는 `create-managed-prefix-list`을 사용하는 방법을 보여 줍니다.

AWS CLI

접두사 목록을 생성하려면

다음 `create-managed-prefix-list` 예제에서는 최대 10개의 항목이 포함된 IPv4 접두사 목록을 생성하고 접두사 목록에 2개의 항목을 생성합니다.

```
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
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon VPC 사용 설명서의 관리형 접두사 목록을 참조하세요.

- 자세한 API 내용은 명령 참조CreateManagedPrefixList의 섹션을 참조하세요. AWS CLI

### `create-nat-gateway`

다음 코드 예시에서는 `create-nat-gateway`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 퍼블릭 NAT 게이트웨이 생성

다음 create-nat-gateway 예제에서는 지정된 서브넷에 퍼블릭 NAT 게이트웨이를 생성하고 탄력적 IP 주소를 지정된 할당 ID와 연결합니다. 퍼블릭 NAT 게이트웨이를 생성할 때 탄력적 IP 주소를 연결해야 합니다.

```
aws ec2 create-nat-gateway \
    --subnet-id subnet-0250c25a1fEXAMPLE \
    --allocation-id eipalloc-09ad461b0dEXAMPLE
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon VPC 사용 설명서의 NAT 게이트웨이를 참조하세요.

예제 2: 프라이빗 NAT 게이트웨이 생성

다음 create-nat-gateway 예제에서는 지정된 서브넷에 프라이빗 NAT 게이트웨이를 생성합니다. 프라이빗 NAT 게이트웨이에는 연결된 탄력적 IP 주소가 없습니다.

```
aws ec2 create-nat-gateway \
    --subnet-id subnet-0250c25a1fEXAMPLE \
    --connectivity-type private
```

출력:

```
{
    "NatGateway": {
```

```
        "CreateTime": "2021-12-01T22:26:00.000Z",
        "NatGatewayAddresses": [
            {}
        ],
        "NatGatewayId": "nat-011b568379EXAMPLE",
        "State": "pending",
        "SubnetId": "subnet-0250c25a1fEXAMPLE",
        "VpcId": "vpc-0a60eb65b4EXAMPLE",
        "ConnectivityType": "private"
    }
}
```

자세한 내용은 Amazon VPC 사용 설명서의 NAT 게이트웨이를 참조하세요.

- 자세한 API 내용은 명령 참조CreateNatGateway의 섹션을 참조하세요. AWS CLI

### `create-network-acl-entry`

다음 코드 예시에서는 `create-network-acl-entry`을 사용하는 방법을 보여 줍니다.

AWS CLI

네트워크 ACL 항목을 생성하려면

이 예제에서는 지정된 네트워크 에 대한 항목을 생성합니다ACL. 규칙은 UDP 포트 53()의 모든 IPv4 주소(0.0.0.0/0DNS)에서 연결된 서브넷으로 트래픽을 수신할 수 있도록 허용합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
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
```

이 예제에서는 TCP 포트 80()의 모든 IPv6 주소(::/0)에서 수신 트래픽을 ACL 허용하는 지정된 네트워크에 대한 규칙을 생성합니다HTTP.

명령:

```
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
```

- 자세한 API 내용은 명령 참조CreateNetworkAclEntry의 섹션을 참조하세요. AWS CLI

### `create-network-acl`

다음 코드 예시에서는 `create-network-acl`을 사용하는 방법을 보여 줍니다.

AWS CLI

네트워크를 생성하려면 ACL

이 예제에서는 지정된 에 ACL 대한 네트워크를 생성합니다VPC.

명령:

```
aws ec2 create-network-acl --vpc-id vpc-a01106c2
```

출력:

```
{
    "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
    }
```

```
}
```

- 자세한 API 내용은 명령 참조CreateNetworkAcl의 섹션을 참조하세요. AWS CLI

### `create-network-insights-access-scope`

다음 코드 예시에서는 `create-network-insights-access-scope`을 사용하는 방법을 보여 줍니다.

AWS CLI

네트워크 액세스 범위를 생성하려면

다음 `create-network-insights-access-scope` 예제에서는 네트워크 액세스 범위를 생성합니다.

```
aws ec2 create-network-insights-access-scope \
    --cli-input-json file://access-scope-file.json
```

`access-scope-file.json`의 콘텐츠:

```
{
    "MatchPaths": [
        {
            "Source": {
                "ResourceStatement": {
                    "Resources": [
                        "vpc-abcd12e3"
                    ]
                }
            }
         }
    ],
    "ExcludePaths": [
        {
            "Source": {
                "ResourceStatement": {
                    "ResourceTypes": [
                        "AWS::EC2::InternetGateway"
                    ]
                }
            }
```

```
        }
    ]
}
```

출력:

```
{
    "NetworkInsightsAccessScope": {
        "NetworkInsightsAccessScopeId": "nis-123456789abc01234",
        "NetworkInsightsAccessScopeArn": "arn:aws:ec2:us-
east-1:123456789012:network-insights-access-scope/nis-123456789abc01234",
        "CreatedDate": "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"
                        ]
                    }
                }
            }
        ]
    }
}
```

자세한 내용은 Network Access Analyzer 가이드의 를 사용하여 AWS CLI Network Access Analyzer 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조CreateNetworkInsightsAccessScope의 섹션을 참조하세요. AWS CLI

### `create-network-insights-path`

다음 코드 예시에서는 `create-network-insights-path`을 사용하는 방법을 보여 줍니다.

AWS CLI

경로를 생성하려면

다음 `create-network-insights-path` 예제에서는 경로를 생성합니다. 소스는 지정된 인터넷 게이트웨이이고 대상은 지정된 EC2 인스턴스입니다. 지정된 프로토콜과 포트를 사용하여 대상에 연결할 수 있는지 확인하려면 `start-network-insights-analysis` 명령을 사용하여 경로를 분석합니다.

```
aws ec2 create-network-insights-path \
    --source igw-0797cccdc9d73b0e5 \
    --destination i-0495d385ad28331c7 \
    --destination-port 22 \
    --protocol TCP
```

출력:

```
{
    "NetworkInsightsPaths": {
        "NetworkInsightsPathId": "nip-0b26f224f1d131fa8",
        "NetworkInsightsPathArn": "arn:aws:ec2:us-east-1:123456789012:network-
insights-path/nip-0b26f224f1d131fa8",
        "CreatedDate": "2021-01-20T22:43:46.933Z",
        "Source": "igw-0797cccdc9d73b0e5",
        "Destination": "i-0495d385ad28331c7",
        "Protocol": "tcp"
    }
}
```

자세한 내용은 Reachability Analyzer 가이드의 시작하기 AWS CLI를 참조하세요.

- 자세한 API 내용은 명령 참조CreateNetworkInsightsPath의 섹션을 참조하세요. AWS CLI

### create-network-interface-permission

다음 코드 예시에서는 create-network-interface-permission을 사용하는 방법을 보여 줍니다.

AWS CLI

네트워크 인터페이스 권한을 생성하려면

이 예제에서는 계정에 네트워크 인터페이스를 인스턴스123456789012에 연결할 eni-1a2b3c4d 수 있는 권한을 부여합니다.

명령:

```
aws ec2 create-network-interface-permission --network-interface-id eni-1a2b3c4d --
aws-account-id 123456789012 --permission INSTANCE-ATTACH
```

출력:

```
{
  "InterfacePermission": {
      "PermissionState": {
          "State": "GRANTED"
      },
      "NetworkInterfacePermissionId": "eni-perm-06fd19020ede149ea",
      "NetworkInterfaceId": "eni-1a2b3c4d",
      "Permission": "INSTANCE-ATTACH",
      "AwsAccountId": "123456789012"
  }
}
```

- 자세한 API 내용은 명령 참조CreateNetworkInterfacePermission의 섹션을 참조하세요. AWS CLI

### create-network-interface

다음 코드 예시에서는 create-network-interface을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 네트워크 인터페이스의 IPv4 주소를 지정하려면

다음 create-network-interface 예제에서는 지정된 기본 IPv4 주소를 사용하여 지정된 서브넷에 대한 네트워크 인터페이스를 생성합니다.

```
aws ec2 create-network-interface \
    --subnet-id subnet-00a24d0d67acf6333 \
    --description "my network interface" \
    --groups sg-09dfba7ed20cda78b \
    --private-ip-address 10.0.8.17
```

출력:

```
{
    "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"
    }
```

```
}
```

예제 2: IPv4 주소와 IPv6 주소를 사용하여 네트워크 인터페이스를 생성하려면

다음 `create-network-interface` 예제에서는 Amazon 에서 선택한 IPv4 주소와 IPv6 주소를 사용하여 지정된 서브넷에 대한 네트워크 인터페이스를 생성합니다EC2.

```
aws ec2 create-network-interface \
    --subnet-id subnet-00a24d0d67acf6333 \
    --description "my dual stack network interface" \
    --ipv6-address-count 1 \
    --groups sg-09dfba7ed20cda78b
```

출력:

```
{
    "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"
    }
}
```

예제 3: 연결 추적 구성 옵션을 사용하여 네트워크 인터페이스를 생성하려면

다음 `create-network-interface` 예제에서는 네트워크 인터페이스를 생성하고 유휴 연결 추적 제한 시간을 구성합니다.

```
aws ec2 create-network-interface \
    --subnet-id subnet-00a24d0d67acf6333 \
    --groups sg-02e57dbcfe0331c1b \
    --connection-tracking-specification TcpEstablishedTimeout=86400,UdpTimeout=60
```

출력:

```
{
    "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"
    }
}
```

예제 4: 탄력적 패브릭 어댑터 생성

다음 `create-network-interface` 예제에서는 를 생성합니다EFA.

```
aws ec2 create-network-interface \
    --interface-type efa \
    --subnet-id subnet-00a24d0d67acf6333 \
    --description "my efa" \
    --groups sg-02e57dbcfe0331c1b
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon EC2 사용 설명서의 탄력적 네트워크 인터페이스를 참조하세요.

- 자세한 API 내용은 명령 참조CreateNetworkInterface의 섹션을 참조하세요. AWS CLI

### `create-placement-group`

다음 코드 예시에서는 `create-placement-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

배치 그룹을 생성하려면

이 예제 명령은 지정된 이름으로 배치 그룹을 생성합니다.

명령:

```
aws ec2 create-placement-group --group-name my-cluster --strategy cluster
```

파티션 배치 그룹을 생성하려면

이 예제 명령은 5개의 파티션이 HDFS-Group-A 있는 라는 파티션 배치 그룹을 생성합니다.

명령:

```
aws ec2 create-placement-group --group-name HDFS-Group-A --strategy partition --
partition-count 5
```

- 자세한 API 내용은 명령 참조CreatePlacementGroup의 섹션을 참조하세요. AWS CLI

### create-replace-root-volume-task

다음 코드 예시에서는 create-replace-root-volume-task을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 루트 볼륨을 초기 시작 상태로 복원하려면

다음 create-replace-root-volume-task 예제에서는 인스턴스 i-0123456789abcdefa의 루트 볼륨을 초기 시작 상태로 복원합니다.

```
aws ec2 create-replace-root-volume-task \
    --instance-id i-0123456789abcdefa
```

출력:

```
{
    "ReplaceRootVolumeTask":
    {
        "InstanceId": "i-0123456789abcdefa",
            "ReplaceRootVolumeTaskId": "replacevol-0111122223333abcd",
            "TaskState": "pending",
            "StartTime": "2022-03-14T15:06:38Z",
            "Tags": []
    }
}
```

자세한 내용은 Amazon Elastic Compute Cloud 사용 설명서의 루트 볼륨 교체를 참조하세요.

예제 2: 루트 볼륨을 특정 스냅샷으로 복원하려면

다음 create-replace-root-volume-task 예제에서는 인스턴스 i-0123456789abcdefa의 루트 볼륨을 스냅샷 snap-0abcdef1234567890로 복원합니다.

```
aws ec2 create-replace-root-volume-task \
```

```
    --instance-id i-0123456789abcdefa \
    --snapshot-id  snap-0abcdef1234567890
```

출력:

```
{
    "ReplaceRootVolumeTask":
    {
        "InstanceId": "i-0123456789abcdefa",
        "ReplaceRootVolumeTaskId": "replacevol-0555566667777abcd",
        "TaskState": "pending",
        "StartTime": "2022-03-14T15:16:28Z",
        "Tags": []
    }
}
```

자세한 내용은 Amazon Elastic Compute Cloud 사용 설명서의 루트 볼륨 교체를 참조하세요.

- 자세한 API 내용은 명령 참조CreateReplaceRootVolumeTask의 섹션을 참조하세요. AWS CLI

### create-reserved-instances-listing

다음 코드 예시에서는 create-reserved-instances-listing을 사용하는 방법을 보여 줍니다.

AWS CLI

예약 인스턴스 마켓플레이스에서 예약 인스턴스를 나열하려면

다음 create-reserved-instances-listing 예제에서는 예약 인스턴스 마켓플레이스에서 지정된 예약 인스턴스에 대한 목록을 생성합니다.

```
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
```

- 자세한 API 내용은 명령 참조CreateReservedInstancesListing의 섹션을 참조하세요. AWS CLI

### create-restore-image-task

다음 코드 예시에서는 create-restore-image-task을 사용하는 방법을 보여 줍니다.

AWS CLI

S3 버킷AMI에서 를 복원하려면

다음 `create-restore-image-task` 예제에서는 S3 버킷AMI에서 를 복원합니다. `describe-store-image-tasks` 출력`S3ObjectKey` `` and ``Bucket`에서 의 값을 사용하고, 의 객체 키AMI와 가 복사AMI된 S3 버킷의 이름을 지정하고, 복원된 의 이름을 지정합니다AMI. 이름은 이 계정의 리전AMIs에서 에 대해 고유해야 합니다. 복원된 는 새 AMI ID를 받게 AMI 됩니다.

```
aws ec2 create-restore-image-task \
    --object-key ami-1234567890abcdef0.bin \
    --bucket my-ami-bucket \
    --name "New AMI Name"
```

출력:

```
{
    "ImageId": "ami-0eab20fe36f83e1a8"
}
```

S3를 AMI 사용하여 를 저장하고 복원하는 방법에 대한 자세한 내용은 Amazon EC2 사용 설명서의 S3 <https://docs.aws.amazon.com/AWS EC2/latest/UserGuide/ami-store-restore.html>을 AMI 사용하여 를 저장하고 복원을 참조하세요.

- 자세한 API 내용은 명령 참조CreateRestoreImageTask의 섹션을 참조하세요. AWS CLI

### `create-route-table`

다음 코드 예시에서는 `create-route-table`을 사용하는 방법을 보여 줍니다.

AWS CLI

라우팅 테이블을 생성하는 방법

이 예제에서는 지정된 에 대한 라우팅 테이블을 생성합니다VPC.

명령:

```
aws ec2 create-route-table --vpc-id vpc-a01106c2
```

출력:

```
{
    "RouteTable": {
        "Associations": [],
        "RouteTableId": "rtb-22574640",
        "VpcId": "vpc-a01106c2",
        "PropagatingVgws": [],
        "Tags": [],
        "Routes": [
            {
                "GatewayId": "local",
                "DestinationCidrBlock": "10.0.0.0/16",
                "State": "active"
            }
        ]
    }
}
```

- 자세한 API 내용은 명령 참조CreateRouteTable의 섹션을 참조하세요. AWS CLI

### `create-route`

다음 코드 예시에서는 `create-route`을 사용하는 방법을 보여 줍니다.

AWS CLI

경로를 생성하려면

이 예제에서는 지정된 라우팅 테이블에 대한 라우팅을 생성합니다. 라우팅은 모든 IPv4 트래픽(`0.0.0.0/0`)과 일치하고 지정된 인터넷 게이트웨이로 라우팅합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 create-route --route-table-id rtb-22574640 --destination-cidr-
block 0.0.0.0/0 --gateway-id igw-c0a643a9
```

이 예제 명령은 route table rtb-g8ff4ea2에 라우팅을 생성합니다. 경로는 IPv4 CIDR 블록 10.0.0.0/16의 트래픽과 일치하고 피VPC어링 연결인 pcx-111aaa22로 라우팅합니다. 이 경로를 사용하면 피어링 연결VPC의 VPC 피어로 트래픽을 전달할 수 있습니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 create-route --route-table-id rtb-g8ff4ea2 --destination-cidr-
block 10.0.0.0/16 --vpc-peering-connection-id pcx-1a2b3c4d
```

이 예제에서는 모든 IPv6 트래픽(::/0)과 일치하는 라우팅을 지정된 라우팅 테이블에 생성하고 지정된 송신 전용 인터넷 게이트웨이로 라우팅합니다.

명령:

```
aws ec2 create-route --route-table-id rtb-dce620b8 --destination-ipv6-cidr-
block ::/0 --egress-only-internet-gateway-id eigw-01eadbd45ecd7943f
```

- 자세한 API 내용은 명령 참조CreateRoute의 섹션을 참조하세요. AWS CLI

### create-security-group

다음 코드 예시에서는 create-security-group을 사용하는 방법을 보여 줍니다.

AWS CLI

EC2-Classic에 대한 보안 그룹을 생성하려면

이 예제에서는 이름이 MySecurityGroup인 보안 그룹을 생성합니다.

명령:

```
aws ec2 create-security-group --group-name MySecurityGroup --description "My
 security group"
```

출력:

```
{
    "GroupId": "sg-903004f8"
}
```

EC2에 대한 보안 그룹을 생성하려면VPC

이 예제에서는 지정된 에 MySecurityGroup 대한 라는 보안 그룹을 생성합니다VPC.

명령:

```
aws ec2 create-security-group --group-name MySecurityGroup --description "My
 security group" --vpc-id vpc-1a2b3c4d
```

출력:

```
{
    "GroupId": "sg-903004f8"
}
```

자세한 내용은 AWS Command Line Interface 사용 설명서의 보안 그룹 사용을 참조하세요.

- 자세한 API 내용은 명령 참조CreateSecurityGroup의 섹션을 참조하세요. AWS CLI

### `create-snapshot`

다음 코드 예시에서는 `create-snapshot`을 사용하는 방법을 보여 줍니다.

AWS CLI

스냅샷을 생성하려면

이 예제 명령은 볼륨 ID가 인 볼륨의 스냅샷`vol-1234567890abcdef0`과 스냅샷을 식별하기 위한 간단한 설명을 생성합니다.

명령:

```
aws ec2 create-snapshot --volume-id vol-1234567890abcdef0 --description "This is my
 root volume snapshot"
```

출력:

```
{
    "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"
```

```
}
```

태그를 사용하여 스냅샷을 생성하려면

이 예제 명령은 스냅샷을 생성하고 purpose=prod 및 costcenter=123이라는 두 개의 태그를 적용합니다.

명령:

```
aws ec2 create-snapshot --volume-id vol-1234567890abcdef0 --description 'Prod
 backup' --tag-specifications 'ResourceType=snapshot,Tags=[{Key=purpose,Value=prod},
{Key=costcenter,Value=123}]'
```

출력:

```
{
    "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"
 }
```

- 자세한 API 내용은 명령 참조CreateSnapshot의 섹션을 참조하세요. AWS CLI

### `create-snapshots`

다음 코드 예시에서는 `create-snapshots`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 다중 볼륨 스냅샷 생성

다음 create-snapshots 예제에서는 지정된 인스턴스에 연결된 모든 볼륨의 스냅샷을 생성합니다.

```
aws ec2 create-snapshots \
    --instance-specification InstanceId=i-1234567890abcdef0 \
    --description "This is snapshot of a volume from my-instance"
```

출력:

```
{
    "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"
        },
        ...
    ]
}
```

예제 2: 소스 볼륨의 태그를 사용하여 다중 볼륨 스냅샷을 생성하려면

다음 create-snapshots 예제에서는 지정된 인스턴스에 연결된 모든 볼륨의 스냅샷을 생성하고 각 볼륨의 태그를 해당 스냅샷으로 복사합니다.

```
aws ec2 create-snapshots \
    --instance-specification InstanceId=i-1234567890abcdef0 \
    --copy-tags-from-source volume \
    --description "This is snapshot of a volume from my-instance"
```

출력:

```
{
    "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"
        }
        ...
    ]
}
```

예제 3: 루트 볼륨을 포함하지 않는 다중 볼륨 스냅샷 생성

다음 create-snapshots 예제에서는 루트 볼륨을 제외하고 지정된 인스턴스에 연결된 모든 볼륨의 스냅샷을 생성합니다.

```
aws ec2 create-snapshots \
    --instance-specification InstanceId=i-1234567890abcdef0,ExcludeBootVolume=true
```

샘플 출력은 예 1을 참조하세요.

예제 4: 다중 볼륨 스냅샷을 생성하고 태그를 추가하려면

다음 create-snapshots 예제에서는 지정된 인스턴스에 연결된 모든 볼륨의 스냅샷을 생성하고 각 스냅샷에 두 개의 태그를 추가합니다.

```
aws ec2 create-snapshots \
    --instance-specification InstanceId=i-1234567890abcdef0 \
    --tag-specifications 'ResourceType=snapshot,Tags=[{Key=Name,Value=backup},
{Key=costcenter,Value=123}]'
```

샘플 출력은 예 1을 참조하세요.

- 자세한 API 내용은 명령 참조CreateSnapshots의 섹션을 참조하세요. AWS CLI

### create-spot-datafeed-subscription

다음 코드 예시에서는 create-spot-datafeed-subscription을 사용하는 방법을 보여 줍니다.

AWS CLI

스팟 인스턴스 데이터 피드를 생성하려면

다음 create-spot-datafeed-subscription 예제에서는 스팟 인스턴스 데이터 피드를 생성합니다.

```
aws ec2 create-spot-datafeed-subscription \
    --bucket my-bucket \
    --prefix spot-data-feed
```

출력:

```
{
    "SpotDatafeedSubscription": {
        "Bucket": "my-bucket",
        "OwnerId": "123456789012",
        "Prefix": "spot-data-feed",
        "State": "Active"
    }
}
```

데이터 피드는 지정한 Amazon S3 버킷에 저장됩니다. 이 데이터 피드의 파일 이름 형식은 다음과 같습니다.

```
my-bucket.s3.amazonaws.com/spot-data-feed/123456789012.YYYY-MM-DD-HH.n.abcd1234.gz
```

자세한 내용은 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 스팟 인스턴스 데이터 피드를 참조하세요.

- 자세한 API 내용은 명령 참조CreateSpotDatafeedSubscription의 섹션을 참조하세요. AWS CLI

### create-store-image-task

다음 코드 예시에서는 create-store-image-task을 사용하는 방법을 보여 줍니다.

AWS CLI

S3 버킷AMI에 를 저장하려면

다음 create-store-image-task 예제에서는 를 S3 버킷AMI에 저장합니다. 의 IDAMI와 를 저장할 S3 버킷의 이름을 지정합니다AMI.

```
aws ec2 create-store-image-task \
  --image-id ami-1234567890abcdef0 \
  --bucket my-ami-bucket
```

출력:

```
{
    "ObjectKey": "ami-1234567890abcdef0.bin"
}
```

자세한 내용은 Amazon EC2 사용 설명서의 S3를 AMI 사용하여 를 저장하고 복원을 참조하세요.

- 자세한 API 내용은 명령 참조CreateStoreImageTask의 섹션을 참조하세요. AWS CLI

### create-subnet-cidr-reservation

다음 코드 예시에서는 create-subnet-cidr-reservation을 사용하는 방법을 보여 줍니다.

AWS CLI

서브넷 CIDR 예약을 생성하려면

다음 create-subnet-cidr-reservation 예제에서는 지정된 서브넷 및 CIDR 범위에 대한 서브넷 CIDR 예약을 생성합니다.

```
aws ec2 create-subnet-cidr-reservation \
    --subnet-id subnet-03c51e2eEXAMPLE \
    --reservation-type prefix \
    --cidr 10.1.0.20/26
```

출력:

```
{
    "SubnetCidrReservation": {
        "SubnetCidrReservationId": "scr-044f977c4eEXAMPLE",
        "SubnetId": "subnet-03c51e2e6cEXAMPLE",
        "Cidr": "10.1.0.16/28",
        "ReservationType": "prefix",
        "OwnerId": "123456789012"
    }
}
```

자세한 내용은 Amazon VPC 사용 설명서의 서브넷 CIDR 예약을 참조하세요.

- 자세한 API 내용은 명령 참조CreateSubnetCidrReservation의 섹션을 참조하세요. AWS CLI

### create-subnet

다음 코드 예시에서는 create-subnet을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: IPv4 CIDR 블록만 사용하여 서브넷 생성

다음 create-subnet 예제에서는 지정된 IPv4 CIDR 블록으로 지정된 에 서브넷VPC을 생성합니다.

```
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}]
```

출력:

```
{
    "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"
    }
}
```

예제 2: IPv4 및 IPv6 CIDR 블록을 모두 사용하여 서브넷 생성

다음 `create-subnet` 예제에서는 지정된 IPv4 및 IPv6 CIDR 블록VPC으로 지정된 에 서브넷을 생성합니다.

```
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}]
```

출력:

```
{
    "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"
    }
}
```

예제 3: IPv6 CIDR 블록만 사용하여 서브넷 생성

다음 `create-subnet` 예제에서는 지정된 IPv6 CIDR 블록으로 지정된 에 서브넷VPC을 생성합니다.

```
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}]
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon VPC 사용 설명서의 VPCs 및 서브넷을 참조하세요.

- 자세한 API 내용은 명령 참조CreateSubnet의 섹션을 참조하세요. AWS CLI

### `create-tags`

다음 코드 예시에서는 `create-tags`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 리소스에 태그를 추가하려면

다음 create-tags 예제에서는 지정된 이미지Stack=production에 태그를 추가하거나 태그 키 AMI가 인 의 기존 태그를 덮어씁니다Stack.

```
aws ec2 create-tags \
    --resources ami-1234567890abcdef0 \
    --tags Key=Stack,Value=production
```

자세한 내용은 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 주제 제목을 참조하세요.

예제 2: 여러 리소스에 태그를 추가하려면

다음 create-tags 예제에서는 AMI 및 인스턴스에 대한 두 개의 태그를 추가(또는 덮어씁니다)합니다. 태그 중 하나에서 키(webserver)는 있지만 값이 없습니다(값이 빈 문자열로 설정됨). 다른 태그에는 키(stack)와 값(Production)이 있습니다.

```
aws ec2 create-tags \
    --resources ami-1a2b3c4d i-1234567890abcdef0 \
    --tags Key=webserver,Value=   Key=stack,Value=Production
```

자세한 내용은 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 주제 제목을 참조하세요.

예제 3: 특수 문자가 포함된 태그를 추가하려면

다음 create-tags 예제에서는 인스턴스에 [Group]=test 태그를 추가합니다. 대괄호([ 및 ])는 이스케이프해야 하는 특수 문자입니다. 다음 예제에서는 각 환경에 적합한 줄 연속 문자도 사용합니다.

Windows를 사용하는 경우 다음과 같이 특수 문자가 있는 요소를 큰따옴표(")로 묶은 다음, 각 큰따옴표 문자 앞에 백슬래시(\)를 붙입니다.

```
aws ec2 create-tags ^
    --resources i-1234567890abcdef0 ^
    --tags Key=\"[Group]\",Value=test
```

Windows 를 사용하는 경우 PowerShell다음과 같이 특수 문자가 있는 요소를 큰따옴표(')로 묶고, 각 큰따옴표 문자 앞에 백슬래시(\)로 묶은 다음 전체 키 및 값 구조를 작은따옴표(')로 묶습니다.

```
aws ec2 create-tags `
```

```
    --resources i-1234567890abcdef0 `
    --tags 'Key=\"[Group]\",Value=test'
```

Linux 또는 OS X를 사용하는 경우 다음과 같이 특수 문자가 있는 요소를 큰따옴표(")로 묶은 다음, 전체 키 및 값 구조를 작음따옴표(')로 묶습니다.

```
aws ec2 create-tags \
    --resources i-1234567890abcdef0 \
    --tags 'Key="[Group]",Value=test'
```

자세한 내용은 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 주제 제목을 참조하세요.

- 자세한 API 내용은 명령 참조CreateTags의 섹션을 참조하세요. AWS CLI

### `create-traffic-mirror-filter-rule`

다음 코드 예시에서는 `create-traffic-mirror-filter-rule`을 사용하는 방법을 보여 줍니다.

AWS CLI

수신 TCP 트래픽에 대한 필터 규칙을 생성하려면

다음 `create-traffic-mirror-filter-rule` 예제에서는 모든 수신 TCP 트래픽을 미러링하는 데 사용할 수 있는 규칙을 생성합니다. 이 명령을 실행하기 전에 `create-traffic-mirror-filter`를 사용하여 트래픽 미러 필터를 생성합니다.

```
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
```

출력:

```
{
    "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"
}
```

자세한 내용은 트래픽 미러링 가이드의 트래픽 미러 필터 생성을 참조하세요. AWS

- 자세한 API 내용은 명령 참조CreateTrafficMirrorFilterRule의 섹션을 참조하세요. AWS CLI

### create-traffic-mirror-filter

다음 코드 예시에서는 create-traffic-mirror-filter을 사용하는 방법을 보여 줍니다.

AWS CLI

트래픽 미러 필터를 생성하려면

다음 create-traffic-mirror-filter 예제에서는 트래픽 미러 필터를 생성합니다. 필터를 생성한 후 create-traffic-mirror-filter-rule를 사용하여 필터에 규칙을 추가합니다.

```
aws ec2 create-traffic-mirror-filter \
    --description "TCP Filter"
```

출력:

```
{
    "ClientToken": "28908518-100b-4987-8233-8c744EXAMPLE",
    "TrafficMirrorFilter": {
        "TrafficMirrorFilterId": "tmf-04812ff784EXAMPLE",
        "Description": "TCP Filter",
        "EgressFilterRules": [],
        "IngressFilterRules": [],
        "Tags": [],
        "NetworkServices": []
    }
```

```
}
```

자세한 내용은 트래픽 미러링 가이드의 트래픽 미러 필터 생성을 참조하세요. AWS

- 자세한 API 내용은 명령 참조CreateTrafficMirrorFilter의 섹션을 참조하세요. AWS CLI

### `create-traffic-mirror-session`

다음 코드 예시에서는 `create-traffic-mirror-session`을 사용하는 방법을 보여 줍니다.

AWS CLI

트래픽 미러 세션을 생성하려면

다음 `create-traffic-mirror-session` 명령은 지정된 소스에 대한 트래픽 미러 세션을 생성하고 패킷의 25바이트에 대한 대상을 생성합니다.

```
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
```

출력:

```
{
    "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"
}
```

자세한 내용은 트래픽 미러링 가이드의 트래픽 미러 세션 생성을 참조하세요. AWS

- 자세한 API 내용은 명령 참조CreateTrafficMirrorSession의 섹션을 참조하세요. AWS CLI

### `create-traffic-mirror-target`

다음 코드 예시에서는 `create-traffic-mirror-target`을 사용하는 방법을 보여 줍니다.

AWS CLI

Network Load Balancer Traffic Mirror 대상을 생성하려면

다음 `create-traffic-mirror-target` 예제에서는 Network Load Balancer Traffic Mirror 대상을 생성합니다.

```
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
```

출력:

```
{
    "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"
}
```

네트워크 트래픽 미러 대상을 생성하려면

다음 `create-traffic-mirror-target` 예제에서는 네트워크 인터페이스 트래픽 미러 대상을 생성합니다.

aws ec2 create-traffic-mirror-target --description “네트워크 인터페이스 대상” --network-interface-id eni-eni-01f6f631eEXAMPLE

출력:

```
{
    "ClientToken": "5289a345-0358-4e62-93d5-47ef3061d65e",
    "TrafficMirrorTarget": {
        "Description": "Network interface target",
        "NetworkInterfaceId": "eni-01f6f631eEXAMPLE",
        "TrafficMirrorTargetId": "tmt-02dcdbe2abEXAMPLE",
        "OwnerId": "111122223333",
        "Type": "network-interface",
        "Tags": []
    }
}
```

자세한 내용은 트래픽 미러링 가이드의 트래픽 미러 대상 생성을 참조하세요. AWS

- 자세한 API 내용은 명령 참조CreateTrafficMirrorTarget의 섹션을 참조하세요. AWS CLI

### `create-transit-gateway-connect-peer`

다음 코드 예시에서는 `create-transit-gateway-connect-peer`을 사용하는 방법을 보여 줍니다.

AWS CLI

Transit Gateway Connect 피어를 생성하려면

다음 `create-transit-gateway-connect-peer` 예제에서는 Connect 피어를 생성합니다.

```
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
```

출력:

```
{
    "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"
                }
            ]
        }
    }
}
```

자세한 내용은 Transit Gateways 가이드의 Transit Gateway Connect 연결 및 Transit Gateway Connect 피어를 참조하세요.

- 자세한 API 내용은 명령 참조CreateTransitGatewayConnectPeer의 섹션을 참조하세요. AWS CLI

### `create-transit-gateway-connect`

다음 코드 예시에서는 `create-transit-gateway-connect`을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 Connect 연결 생성

다음 `create-transit-gateway-connect` 예제에서는 지정된 연결에 대해 'gre' 프로토콜을 사용하여 Connect 연결을 생성합니다.

```
aws ec2 create-transit-gateway-connect \
    --transport-transit-gateway-attachment-id tgw-attach-0a89069f57EXAMPLE \
    --options "Protocol=gre"
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 Transit Gateways 가이드의 Transit Gateway Connect 연결 및 Transit Gateway Connect 피어를 참조하세요.

- 자세한 API 내용은 명령 참조CreateTransitGatewayConnect의 섹션을 참조하세요. AWS CLI

### create-transit-gateway-multicast-domain

다음 코드 예시에서는 create-transit-gateway-multicast-domain을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: IGMP 멀티캐스트 도메인 생성

다음 create-transit-gateway-multicast-domain 예제에서는 지정된 전송 게이트웨이에 대한 멀티캐스트 도메인을 생성합니다. 정적 소스가 비활성화된 경우 멀티캐스트 도메인과 연결된 서브넷의 모든 인스턴스가 멀티캐스트 트래픽을 전송할 수 있습니다. 하나 이상의 멤버가 IGMP 프로토콜을 사용하는 경우 IGMPv2 지원을 활성화해야 합니다.

```
aws ec2 create-transit-gateway-multicast-domain \
    --transit-gateway-id tgw-0bf0bffefaEXAMPLE \
    --options StaticSourcesSupport=disable,Igmpv2Support=enable
```

출력:

```
{
    "TransitGatewayMulticastDomain": {
        "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c9e29e2a7EXAMPLE",
        "TransitGatewayId": "tgw-0bf0bffefaEXAMPLE",
        "TransitGatewayMulticastDomainArn": "arn:aws:ec2:us-
west-2:123456789012:transit-gateway-multicast-domain/tgw-mcast-
domain-0c9e29e2a7EXAMPLE",
        "OwnerId": "123456789012",
        "Options": {
            "Igmpv2Support": "enable",
            "StaticSourcesSupport": "disable",
            "AutoAcceptSharedAssociations": "disable"
        },
        "State": "pending",
        "CreationTime": "2021-09-29T22:17:13.000Z"
    }
}
```

예제 2: 정적 멀티캐스트 도메인 생성

다음 `create-transit-gateway-multicast-domain` 예제에서는 지정된 전송 게이트웨이에 대한 멀티캐스트 도메인을 생성합니다. 정적 소스가 활성화된 경우 소스를 정적 방식으로 추가해야 합니다.

```
aws ec2 create-transit-gateway-multicast-domain \
    --transit-gateway-id tgw-0bf0bffefaEXAMPLE \
    --options StaticSourcesSupport=enable,Igmpv2Support=disable
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Transit Gateways 가이드의 멀티캐스트 도메인 관리를 참조하세요.

- 자세한 API 내용은 명령 참조CreateTransitGatewayMulticastDomain의 섹션을 참조하세요. AWS CLI

### `create-transit-gateway-peering-attachment`

다음 코드 예시에서는 `create-transit-gateway-peering-attachment`을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 피어링 연결 생성

다음 `create-transit-gateway-peering-attachment` 예제에서는 지정된 두 전송 게이트웨이 간에 피어링 연결 요청을 생성합니다.

```
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
```

출력:

```
{
    "TransitGatewayPeeringAttachment": {
        "TransitGatewayAttachmentId": "tgw-attach-4455667788aabbccd",
        "RequesterTgwInfo": {
            "TransitGatewayId": "tgw-123abc05e04123abc",
            "OwnerId": "123456789012",
            "Region": "us-west-2"
        },
        "AccepterTgwInfo": {
```

```
            "TransitGatewayId": "tgw-11223344aabbcc112",
            "OwnerId": "123456789012",
            "Region": "us-east-2"
        },
        "State": "initiatingRequest",
        "CreationTime": "2019-12-09T11:38:05.000Z"
    }
}
```

자세한 내용은 Transit Gateways 가이드의 Transit Gateway 피어링 연결을 참조하세요.

- 자세한 API 내용은 명령 참조CreateTransitGatewayPeeringAttachment의 섹션을 참조하세요. AWS CLI

### `create-transit-gateway-policy-table`

다음 코드 예시에서는 `create-transit-gateway-policy-table`을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 정책 테이블을 생성하려면

다음 `create-transit-gateway-policy-table` 예제에서는 지정된 전송 게이트웨이에 대한 전송 게이트웨이 정책 테이블을 생성합니다.

```
aws ec2 create-transit-gateway-policy-table \
    --transit-gateway-id tgw-067f8505c18f0bd6e
```

출력:

```
{
    "TransitGatewayPolicyTable": {
        "TransitGatewayPolicyTableId": "tgw-ptb-0a16f134b78668a81",
        "TransitGatewayId": "tgw-067f8505c18f0bd6e",
        "State": "pending",
        "CreationTime": "2023-11-28T16:36:43+00:00"
    }
}
```

자세한 내용은 Transit Gateway 사용 설명서의 Transit Gateway 정책 테이블을 참조하세요.

- 자세한 API 내용은 명령 참조CreateTransitGatewayPolicyTable의 섹션을 참조하세요. AWS CLI

### create-transit-gateway-prefix-list-reference

다음 코드 예시에서는 create-transit-gateway-prefix-list-reference을 사용하는 방법을 보여 줍니다.

AWS CLI

접두사 목록에 대한 참조를 생성하려면

다음 create-transit-gateway-prefix-list-reference 예제에서는 지정된 전송 게이트웨이 라우팅 테이블에 지정된 접두사 목록에 대한 참조를 생성합니다.

```
aws ec2 create-transit-gateway-prefix-list-reference \
    --transit-gateway-route-table-id tgw-rtb-0123456789abcd123 \
    --prefix-list-id pl-1111122222222333 \
    --transit-gateway-attachment-id tgw-attach-aaaaaabbbbbb11111
```

출력:

```
{
    "TransitGatewayPrefixListReference": {
        "TransitGatewayRouteTableId": "tgw-rtb-0123456789abcd123",
        "PrefixListId": "pl-11111122222222333",
        "PrefixListOwnerId": "123456789012",
        "State": "pending",
        "Blackhole": false,
        "TransitGatewayAttachment": {
            "TransitGatewayAttachmentId": "tgw-attach-aaaaaabbbbbb11111",
            "ResourceType": "vpc",
            "ResourceId": "vpc-112233445566aabbc"
        }
    }
}
```

자세한 내용은 Transit Gateways 가이드의 접두사 목록 참조를 참조하세요.

- 자세한 API 내용은 명령 참조CreateTransitGatewayPrefixListReference의 섹션을 참조하세요. AWS CLI

### create-transit-gateway-route-table

다음 코드 예시에서는 create-transit-gateway-route-table을 사용하는 방법을 보여 줍니다.

AWS CLI

Transit Gateway Route Table을 생성하려면

다음 `create-transit-gateway-route-table` 예제에서는 지정된 전송 게이트웨이에 대한 라우팅 테이블을 생성합니다.

```
aws ec2 create-transit-gateway-route-table \
    --transit-gateway-id tgw-0262a0e521EXAMPLE
```

출력:

```
{
    "TransitGatewayRouteTable": {
        "TransitGatewayRouteTableId": "tgw-rtb-0960981be7EXAMPLE",
        "TransitGatewayId": "tgw-0262a0e521EXAMPLE",
        "State": "pending",
        "DefaultAssociationRouteTable": false,
        "DefaultPropagationRouteTable": false,
        "CreationTime": "2019-07-10T19:01:46.000Z"
    }
}
```

자세한 내용은 Transit Gateways 가이드의 전송 게이트웨이 라우팅 테이블 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateTransitGatewayRouteTable의 섹션을 참조하세요. AWS CLI

### `create-transit-gateway-route`

다음 코드 예시에서는 `create-transit-gateway-route`을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 경로를 생성하려면

다음 `create-transit-gateway-route` 예제에서는 지정된 라우팅 테이블에 대해 지정된 대상으로 라우팅을 생성합니다.

```
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
```

출력:

```
{
    "Route": {
        "DestinationCidrBlock": "10.0.2.0/24",
        "TransitGatewayAttachments": [
            {
                "ResourceId": "vpc-0065acced4EXAMPLE",
                "TransitGatewayAttachmentId": "tgw-attach-0b5968d3b6EXAMPLE",
                "ResourceType": "vpc"
            }
        ],
        "Type": "static",
        "State": "active"
    }
}
```

자세한 내용은 Transit Gateways 가이드의 Transit Gateway route table을 참조하세요.

- 자세한 API 내용은 명령 참조CreateTransitGatewayRoute의 섹션을 참조하세요. AWS CLI

### `create-transit-gateway-vpc-attachment`

다음 코드 예시에서는 `create-transit-gateway-vpc-attachment`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 전송 게이트웨이를 에 연결하려면 VPC

다음 `create-transit-gateway-vpc-attachment` 예제에서는 지정된 에 대한 전송 게이트웨이 연결을 생성합니다VPC.

```
aws ec2 create-transit-gateway-vpc-attachment \
    --transit-gateway-id tgw-0262a0e521EXAMPLE \
    --vpc-id vpc-07e8ffd50f49335df \
    --subnet-id subnet-0752213d59EXAMPLE
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 Transit Gateways 가이드의 에 대한 전송 게이트웨이 연결 생성을 VPC 참조하세요.

예제 2: 전송 게이트웨이를 의 여러 서브넷에 연결하려면 VPC

다음 `create-transit-gateway-vpc-attachment` 예제에서는 지정된 VPC 및 서브넷에 대한 전송 게이트웨이 연결을 생성합니다.

```
aws ec2 create-transit-gateway-vpc-attachment \
    --transit-gateway-id tgw-02f776b1a7EXAMPLE  \
    --vpc-id vpc-3EXAMPLE \
    --subnet-ids "subnet-dEXAMPLE" "subnet-6EXAMPLE"
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 Transit Gateways 가이드의 에 대한 전송 게이트웨이 연결 생성을 VPC 참조하세요.

- 자세한 API 내용은 명령 참조CreateTransitGatewayVpcAttachment의 섹션을 참조하세요. AWS CLI

### `create-transit-gateway`

다음 코드 예시에서는 `create-transit-gateway`을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이를 생성하려면

다음 `create-transit-gateway` 예제에서는 전송 게이트웨이를 생성합니다.

```
aws ec2 create-transit-gateway \
    --description MyTGW \
    --
options AmazonSideAsn=64516,AutoAcceptSharedAttachments=enable,DefaultRouteTableAssociation=
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 Transit Gateways 가이드의 전송 게이트웨이 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateTransitGateway의 섹션을 참조하세요. AWS CLI

### `create-verified-access-endpoint`

다음 코드 예시에서는 `create-verified-access-endpoint`을 사용하는 방법을 보여 줍니다.

AWS CLI

Verified Access 엔드포인트를 생성하려면

다음 `create-verified-access-endpoint` 예제에서는 지정된 Verified Access 그룹에 대한 Verified Access 엔드포인트를 생성합니다. 지정된 네트워크 인터페이스와 보안 그룹은 동일한 에 속해야 합니다VPC.

```
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}]
```

출력:

```
{
    "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"
            }
        ]
    }
}
```

자세한 내용은 Verified Access 사용 설명서의 Verified Access 엔드포인트를 참조하세요. AWS

- 자세한 API 내용은 명령 참조CreateVerifiedAccessEndpoint의 섹션을 참조하세요. AWS CLI

### `create-verified-access-group`

다음 코드 예시에서는 `create-verified-access-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

Verified Access 그룹을 생성하려면

다음 create-verified-access-group 예제에서는 지정된 Verified Access 인스턴스에 대한 Verified Access 그룹을 생성합니다.

```
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}]
```

출력:

```
{
    "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"
            }
        ]
    }
}
```

자세한 내용은 Verified Access 사용 설명서의 Verified Access 그룹을 참조하세요. AWS

- 자세한 API 내용은 명령 참조CreateVerifiedAccessGroup의 섹션을 참조하세요. AWS CLI

### create-verified-access-instance

다음 코드 예시에서는 create-verified-access-instance을 사용하는 방법을 보여 줍니다.

AWS CLI

Verified Access 인스턴스를 생성하려면

다음 create-verified-access-instance 예제에서는 이름 태그가 있는 Verified Access 인스턴스를 생성합니다.

```
aws ec2 create-verified-access-instance \
    --tag-specifications ResourceType=verified-access-
instance,Tags=[{Key=Name,Value=my-va-instance}]
```

출력:

```
{
    "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"
            }
        ]
    }
}
```

자세한 내용은 Verified Access 사용 설명서의 Verified Access 인스턴스를 참조하세요. AWS

- 자세한 API 내용은 명령 참조CreateVerifiedAccessInstance의 섹션을 참조하세요. AWS CLI

### create-verified-access-trust-provider

다음 코드 예시에서는 create-verified-access-trust-provider을 사용하는 방법을 보여 줍니다.

AWS CLI

Verified Access 신뢰 공급자를 생성하려면

다음 create-verified-access-trust-provider 예제에서는 AWS Identity Center를 사용하여 Verified Access 신뢰 공급자를 설정합니다.

```
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}]
```

출력:

```
{
    "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"
            }
        ]
    }
}
```

자세한 내용은 Verified Access 사용 설명서의 Verified Access에 대한 신뢰 공급자를 참조하세요. AWS

- 자세한 API 내용은 명령 참조CreateVerifiedAccessTrustProvider의 섹션을 참조하세요. AWS CLI

### `create-volume`

다음 코드 예시에서는 `create-volume`을 사용하는 방법을 보여 줍니다.

AWS CLI

비어 있는 범용SSD(gp2) 볼륨을 생성하려면

다음 `create-volume` 예제에서는 지정된 가용 영역에 80GiB 범용SSD(gp2) 볼륨을 생성합니다. 현재 리전은 여야 합니다. `us-east-1`또는 `--region` 파라미터를 추가하여 명령의 리전을 지정할 수 있습니다.

```
aws ec2 create-volume \
```

```
    --volume-type gp2 \
    --size 80 \
    --availability-zone us-east-1a
```

출력:

```
{
    "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
}
```

볼륨 유형을 지정하지 않으면 기본 볼륨 유형은 입니다gp2.

```
aws ec2 create-volume \
    --size 80 \
    --availability-zone us-east-1a
```

예제 2: 스냅샷에서 프로비저닝된 IOPS SSD (io1) 볼륨 생성

다음 `create-volume` 예제에서는 지정된 스냅샷을 사용하여 IOPS 지정된 가용 영역에 1,000개의 프로비저닝된 IOPS SSD (io1) 볼륨을 생성합니다.

```
aws ec2 create-volume \
    --volume-type io1 \
    --iops 1000 \
    --snapshot-id snap-066877671789bd71b \
    --availability-zone us-east-1a
```

출력:

```
{
    "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
}
```

예제 3: 암호화된 볼륨 생성

다음 create-volume 예제에서는 EBS 암호화 기본값을 사용하여 암호화된 볼륨CMK을 생성합니다. 기본적으로 암호화가 비활성화된 경우 다음과 같이 --encrypted 파라미터를 지정해야 합니다.

```
aws ec2 create-volume \
    --size 80 \
    --encrypted \
    --availability-zone us-east-1a
```

출력:

```
{
    "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
}
```

기본적으로 암호화가 활성화된 경우 다음 예제 명령은 --encrypted 파라미터 없이도 암호화된 볼륨을 생성합니다.

```
aws ec2 create-volume \
    --size 80 \
    --availability-zone us-east-1a
```

--kms-key-id 파라미터를 사용하여 고객 관리형 를 지정하는 경우 기본적으로 암호화가 활성화되어 있더라도 --encrypted 파라미터를 지정CMK해야 합니다.

```
aws ec2 create-volume \
    --volume-type gp2 \
    --size 80 \
    --encrypted \
    --kms-key-id 0ea3fef3-80a7-4778-9d8c-1c0c6EXAMPLE \
    --availability-zone us-east-1a
```

예제 4: 태그를 사용하여 볼륨을 생성하려면

다음 create-volume 예제에서는 볼륨을 생성하고 두 개의 태그를 추가합니다.

```
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}]'
```

- 자세한 API 내용은 명령 참조CreateVolume의 섹션을 참조하세요. AWS CLI

### create-vpc-endpoint-connection-notification

다음 코드 예시에서는 create-vpc-endpoint-connection-notification을 사용하는 방법을 보여 줍니다.

AWS CLI

엔드포인트 연결 알림을 생성하려면

이 예제에서는 인터페이스 엔드포인트가 서비스에 연결되었을 때와 엔드포인트가 서비스에 수락되었을 때 알림을 보내는 특정 엔드포인트 서비스에 대한 알림을 생성합니다.

명령:

```
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
```

출력:

```
{
   "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"
   }
 }
```

- 자세한 API 내용은 명령 참조CreateVpcEndpointConnectionNotification의 섹션을 참조하세요. AWS CLI

### `create-vpc-endpoint-service-configuration`

다음 코드 예시에서는 `create-vpc-endpoint-service-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 인터페이스 엔드포인트에 대한 엔드포인트 서비스 구성을 생성하려면

다음 `create-vpc-endpoint-service-configuration` 예제에서는 Network Load Balancer를 사용하여 VPC 엔드포인트 서비스 구성을 생성합니다`nlb-vpce`. 또한 이 예제에서는 인터페이스 엔드포인트를 통해 서비스에 연결하기 위한 요청을 수락하도록 지정합니다.

```
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
```

출력:

```
{
```

```
    "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"
        ]
    }
}
```

예제 2: Gateway Load Balancer 엔드포인트에 대한 엔드포인트 서비스 구성을 생성하려면

다음 `create-vpc-endpoint-service-configuration` 예제에서는 Gateway Load Balancer를 사용하여 VPC 엔드포인트 서비스 구성을 생성합니다`GWLBService`. Gateway Load Balancer 엔드포인트를 통해 서비스에 연결하라는 요청은 자동으로 수락됩니다.

```
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
```

출력:

```
{
    "ServiceConfiguration": {
        "ServiceType": [
            {
                "ServiceType": "GatewayLoadBalancer"
            }
```

```
        ],
        "ServiceId": "vpce-svc-123123a1c43abc123",
        "ServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-123123a1c43abc123",
        "ServiceState": "Available",
        "AvailabilityZones": [
            "us-east-1d"
        ],
        "AcceptanceRequired": false,
        "ManagesVpcEndpoints": false,
        "GatewayLoadBalancerArns": [
            "arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/gwy/
GWLBService/123123123123abcc"
        ]
    }
}
```

자세한 내용은 Amazon VPC 사용 설명서의 VPC 엔드포인트 서비스를 참조하세요.

- 자세한 API 내용은 명령 참조CreateVpcEndpointServiceConfiguration의 섹션을 참조하세요. AWS CLI

### `create-vpc-endpoint`

다음 코드 예시에서는 `create-vpc-endpoint`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 게이트웨이 엔드포인트 생성

다음 `create-vpc-endpoint` 예제에서는 `us-east-1` 리전에서 VPC `vpc-1a2b3c4d` 및 Amazon S3 간에 게이트웨이 VPC 엔드포인트를 생성하고 라우팅 테이블을 엔드포인트`rtb-11aa22bb`와 연결합니다.

```
aws ec2 create-vpc-endpoint \
    --vpc-id vpc-1a2b3c4d \
    --service-name com.amazonaws.us-east-1.s3 \
    --route-table-ids rtb-11aa22bb
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS PrivateLink 가이드의 게이트웨이 엔드포인트 생성을 참조하세요.

예제 2: 인터페이스 엔드포인트 생성

다음 `create-vpc-endpoint` 예제에서는 `us-east-1` 리전에서 VPC `vpc-1a2b3c4d` 및 Amazon S3 간에 인터페이스 VPC 엔드포인트를 생성합니다. 이 명령은 서브넷 에서 엔드포인트를 생성하고`subnet-1a2b3c4d`, 이를 보안 그룹 에 연결하고`sg-1a2b3c4d`, “서비스” 키와 “S3” 값을 가진 태그를 추가합니다.

```
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}]
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 사용 설명서의 인터페이스 엔드포인트 생성을 참조하세요. AWS PrivateLink

예제 3: Gateway Load Balancer 엔드포인트 생성

다음 `create-vpc-endpoint` 예제에서는 VPC `vpc-111122223333aabbc` 및 사이에 Gateway Load Balancer 를 사용하여 구성된 서비스와 Gateway Load Balancer 엔드포인트를 생성합니다.

```
aws ec2 create-vpc-endpoint \
```

```
    --service-name com.amazonaws.vpce.us-east-1.vpce-svc-123123a1c43abc123 \
    --vpc-endpoint-type GatewayLoadBalancer \
    --vpc-id vpc-111122223333aabbc \
    --subnet-ids subnet-0011aabbcc2233445
```

출력:

```
{
    "VpcEndpoint": {
        "VpcEndpointId": "vpce-aabbaabbaabbaabba",
        "VpcEndpointType": "GatewayLoadBalancer",
        "VpcId": "vpc-111122223333aabbc",
        "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"
    }
}
```

자세한 내용은 사용 설명서의 Gateway Load Balancer 엔드포인트를 참조하세요. AWS PrivateLink

- 자세한 API 내용은 명령 참조CreateVpcEndpoint의 섹션을 참조하세요. AWS CLI

### `create-vpc-peering-connection`

다음 코드 예시에서는 `create-vpc-peering-connection`을 사용하는 방법을 보여 줍니다.

AWS CLI

간에 VPC 피어링 연결을 생성하려면 VPCs

이 예제에서는 VPCs vpc-1a2b3c4d와 vpc-11122233 간의 피어링 연결을 요청합니다.

명령:

```
aws ec2 create-vpc-peering-connection --vpc-id vpc-1a2b3c4d --peer-vpc-
id vpc-11122233
```

출력:

```
{
    "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"
        }
    }
}
```

다른 계정VPC에서 와 VPC 피어링 연결을 생성하려면

이 예제에서는 VPC (vpc-1a2b3c4d)와 AWS 계정 123456789012에 속하는 VPC (vpc-11122233) 간의 피어링 연결을 요청합니다.

명령:

```
aws ec2 create-vpc-peering-connection --vpc-id vpc-1a2b3c4d --peer-vpc-
id vpc-11122233 --peer-owner-id 123456789012
```

다른 리전VPC에서 와 VPC 피어링 연결을 생성하려면

이 예제에서는 현재 리전(vpc-1a2b3c4d)VPC의 와 `us-west-2` 리전의 계정VPC(vpc-11122233) 간에 피어링 연결을 요청합니다.

명령:

```
aws ec2 create-vpc-peering-connection --vpc-id vpc-1a2b3c4d --peer-vpc-
id vpc-11122233 --peer-region us-west-2
```

이 예제에서는 현재 리전(vpc-1a2b3c4d)VPC의 와 us-west-2 리전에 있는 AWS 계정 123456789012에 속하는 VPC (vpc-11122233) 간의 피어링 연결을 요청합니다.

명령:

```
aws ec2 create-vpc-peering-connection --vpc-id vpc-1a2b3c4d --peer-vpc-
id vpc-11122233 --peer-owner-id 123456789012 --peer-region us-west-2
```

- 자세한 API 내용은 명령 참조CreateVpcPeeringConnection의 섹션을 참조하세요. AWS CLI

### create-vpc

다음 코드 예시에서는 create-vpc을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 생성 VPC

다음 create-vpc 예제에서는 지정된 IPv4 CIDR 블록과 이름 태그VPC로 를 생성합니다.

```
aws ec2 create-vpc \
    --cidr-block 10.0.0.0/16 \
    --tag-specifications ResourceType=vpc,Tags=[{Key=Name,Value=MyVpc}]
```

출력:

```
{
    "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"
            }
        ]
    }
}
```

예제 2: 전용 테넌시VPC를 사용하여 를 생성하려면

다음 `create-vpc` 예제에서는 지정된 IPv4 CIDR 블록과 전용 테넌시를 VPC 사용하여 를 생성합니다.

```
aws ec2 create-vpc \
    --cidr-block 10.0.0.0/16 \
    --instance-tenancy dedicated
```

출력:

```
{
    "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
    }
}
```

예제 3: IPv6 CIDR 블록VPC을 사용하여 를 생성하려면

다음 `create-vpc` 예제에서는 Amazon에서 제공하는 IPv6 CIDR 블록VPC을 사용하여 를 생성합니다.

```
aws ec2 create-vpc \
    --cidr-block 10.0.0.0/16 \
    --amazon-provided-ipv6-cidr-block
```

출력:

```
{
    "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
    }
}
```

예제 4: IPAM 풀CIDR에서 VPC를 사용하여 를 생성하려면

다음 `create-vpc` 예제에서는 Amazon VPC IP Address Manager(IPAM) 풀CIDR에서 VPC를 사용하여 를 생성합니다.

Linux 및 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"}]
```

출력:

```
{
    "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"
            }
        ]
    }
}
```

자세한 내용은 Amazon VPC IPAM 사용 설명서의 IPAM 풀을 VPC 사용하는 생성 CIDR 섹션을 참조하세요.

- 자세한 API 내용은 명령 참조CreateVpc의 섹션을 참조하세요. AWS CLI

### `create-vpn-connection-route`

다음 코드 예시에서는 `create-vpn-connection-route`을 사용하는 방법을 보여 줍니다.

AWS CLI

VPN 연결에 대한 정적 경로를 생성하려면

이 예제에서는 지정된 VPN 연결에 대한 정적 경로를 생성합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 create-vpn-connection-route --vpn-connection-id vpn-40f41529 --destination-cidr-block 11.12.0.0/16
```

- 자세한 API 내용은 명령 참조CreateVpnConnectionRoute의 섹션을 참조하세요. AWS CLI

### create-vpn-connection

다음 코드 예시에서는 create-vpn-connection을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 동적 라우팅으로 VPN 연결 생성

다음 create-vpn-connection 예제에서는 지정된 가상 프라이빗 게이트웨이와 지정된 고객 게이트웨이 간에 VPN 연결을 생성하고 VPN 연결에 태그를 적용합니다. 출력에는 XML 형식의 고객 게이트웨이 디바이스에 대한 구성 정보가 포함됩니다.

```
aws ec2 create-vpn-connection \
    --type ipsec.1 \
    --customer-gateway-id cgw-001122334455aabbc \
    --vpn-gateway-id vgw-1a1a1a1a1a1a2b2b2 \
    --tag-specification 'ResourceType=vpn-connection,Tags=[{Key=Name,Value=BGP-
VPN}]'
```

출력:

```
{
    "VpnConnection": {
        "CustomerGatewayConfiguration": "...configuration information...",
        "CustomerGatewayId": "cgw-001122334455aabbc",
        "Category": "VPN",
        "State": "pending",
        "VpnConnectionId": "vpn-123123123123abcab",
        "VpnGatewayId": "vgw-1a1a1a1a1a1a2b2b2",
        "Options": {
            "EnableAcceleration": false,
            "StaticRoutesOnly": false,
            "LocalIpv4NetworkCidr": "0.0.0.0/0",
            "RemoteIpv4NetworkCidr": "0.0.0.0/0",
            "TunnelInsideIpVersion": "ipv4",
            "TunnelOptions": [
                {},
                {}
            ]
        },
        "Routes": [],
        "Tags": [
```

```
            {
                "Key": "Name",
                "Value": "BGP-VPN"
            }
        ]
    }
}
```

자세한 내용은 AWS Site-to-Site VPN 사용 설명서의 작동 방식을 AWS Site-to-Site VPN 참조하세요.

예제 2: 정적 라우팅으로 VPN 연결 생성

다음 `create-vpn-connection` 예제에서는 지정된 가상 프라이빗 게이트웨이와 지정된 고객 게이트웨이 간에 VPN 연결을 생성합니다. 옵션은 정적 라우팅을 지정합니다. 출력에는 고객 게이트웨이 디바이스의 구성 정보가 XML 형식으로 포함됩니다.

```
aws ec2 create-vpn-connection \
    --type ipsec.1 \
    --customer-gateway-id cgw-001122334455aabbc \
    --vpn-gateway-id vgw-1a1a1a1a1a1a2b2b2 \
    --options "{\"StaticRoutesOnly\":true}"
```

출력:

```
{
    "VpnConnection": {
        "CustomerGatewayConfiguration": "..configuration information...",
        "CustomerGatewayId": "cgw-001122334455aabbc",
        "Category": "VPN",
        "State": "pending",
        "VpnConnectionId": "vpn-123123123123abcab",
        "VpnGatewayId": "vgw-1a1a1a1a1a1a2b2b2",
        "Options": {
            "EnableAcceleration": false,
            "StaticRoutesOnly": true,
            "LocalIpv4NetworkCidr": "0.0.0.0/0",
            "RemoteIpv4NetworkCidr": "0.0.0.0/0",
            "TunnelInsideIpVersion": "ipv4",
            "TunnelOptions": [
                {},
                {}
```

```
            ]
        },
        "Routes": [],
        "Tags": []
    }
}
```

자세한 내용은 AWS Site-to-Site VPN 사용 설명서의 작동 방식을 AWS Site-to-Site VPN 참조하세요.

예제 3: VPN 연결을 생성하고 내부 키와 CIDR 미리 공유한 키를 지정하려면

다음 `create-vpn-connection` 예제에서는 VPN 연결을 생성하고 각 터널에 대한 내부 IP 주소 CIDR 블록과 사용자 지정 사전 공유 키를 지정합니다. 지정된 값이 `CustomerGatewayConfiguration` 정보에 반환됩니다.

```
aws ec2 create-vpn-connection \
    --type ipsec.1 \
    --customer-gateway-id cgw-001122334455aabbc \
    --vpn-gateway-id vgw-1a1a1a1a1a1a2b2b2 \
    --options
 TunnelOptions='[{TunnelInsideCidr=169.254.12.0/30,PreSharedKey=ExamplePreSharedKey1},
{TunnelInsideCidr=169.254.13.0/30,PreSharedKey=ExamplePreSharedKey2}]'
```

출력:

```
{
    "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": []
    }
}
```

자세한 내용은 AWS Site-to-Site VPN 사용 설명서의 작동 방식을 AWS Site-to-Site VPN 참조하세요.

예제 4: IPv6 트래픽을 지원하는 VPN 연결을 생성하려면

다음 `create-vpn-connection` 예제에서는 지정된 전송 게이트웨이와 지정된 고객 게이트웨이 간의 IPv6 트래픽을 지원하는 VPN 연결을 생성합니다. 두 터널의 터널 옵션은 IKE 가 협상을 시작해야 AWS 한다고 지정합니다.

```
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}]
```

출력:

```
{
    "VpnConnection": {
        "CustomerGatewayConfiguration": "..configuration information...",
        "CustomerGatewayId": "cgw-001122334455aabbc",
        "Category": "VPN",
        "State": "pending",
        "VpnConnectionId": "vpn-11111111122222222",
        "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": []
    }
}
```

자세한 내용은 AWS Site-to-Site VPN 사용 설명서의 작동 방식을 AWS Site-to-Site VPN 참조하세요.

- 자세한 API 내용은 명령 참조CreateVpnConnection의 섹션을 참조하세요. AWS CLI

### `create-vpn-gateway`

다음 코드 예시에서는 `create-vpn-gateway`을 사용하는 방법을 보여 줍니다.

AWS CLI

가상 프라이빗 게이트웨이를 생성하려면

이 예제에서는 가상 프라이빗 게이트웨이를 생성합니다.

명령:

```
aws ec2 create-vpn-gateway --type ipsec.1
```

출력:

```
{
    "VpnGateway": {
        "AmazonSideAsn": 64512,
        "State": "available",
        "Type": "ipsec.1",
        "VpnGatewayId": "vgw-9a4cacf3",
        "VpcAttachments": []
    }
}
```

특정 Amazon 측을 사용하여 가상 프라이빗 게이트웨이를 생성하려면 ASN

이 예제에서는 가상 프라이빗 게이트웨이를 생성하고 BGP 세션의 Amazon 측에 대한 Autonomous System Number(ASN)를 지정합니다.

명령:

```
aws ec2 create-vpn-gateway --type ipsec.1 --amazon-side-asn 65001
```

출력:

```
{
    "VpnGateway": {
        "AmazonSideAsn": 65001,
        "State": "available",
        "Type": "ipsec.1",
        "VpnGatewayId": "vgw-9a4cacf3",
        "VpcAttachments": []
    }
}
```

- 자세한 API 내용은 명령 참조CreateVpnGateway의 섹션을 참조하세요. AWS CLI

### `delete-carrier-gateway`

다음 코드 예시에서는 `delete-carrier-gateway`을 사용하는 방법을 보여 줍니다.

AWS CLI

캐리어 게이트웨이를 삭제하려면

다음 delete-carrier-gateway 예제에서는 지정된 캐리어 게이트웨이를 삭제합니다.

```
aws ec2 delete-carrier-gateway \
    --carrier-gateway-id cagw-0465cdEXAMPLE1111
```

출력:

```
{
    "CarrierGateway": {
        "CarrierGatewayId": "cagw-0465cdEXAMPLE1111",
        "VpcId": "vpc-0c529aEXAMPLE1111",
        "State": "deleting",
        "OwnerId": "123456789012"
    }
}
```

자세한 내용은 Amazon Virtual Private Cloud 사용 설명서의 통신 사업자 게이트웨이를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteCarrierGateway의 섹션을 참조하세요. AWS CLI

### delete-client-vpn-endpoint

다음 코드 예시에서는 delete-client-vpn-endpoint을 사용하는 방법을 보여 줍니다.

AWS CLI

클라이언트 VPN 엔드포인트를 삭제하려면

다음 delete-client-vpn-endpoint 예제에서는 지정된 클라이언트 VPN 엔드포인트를 삭제합니다.

```
aws ec2 delete-client-vpn-endpoint \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

출력:

```
{
    "Status": {
        "Code": "deleting"
    }
```

```
}
```

자세한 내용은 클라이언트 관리자 안내서의 클라이언트 VPN 엔드포인트를 참조하세요. AWS VPN

- 자세한 API 내용은 명령 참조DeleteClientVpnEndpoint의 섹션을 참조하세요. AWS CLI

### delete-client-vpn-route

다음 코드 예시에서는 delete-client-vpn-route을 사용하는 방법을 보여 줍니다.

AWS CLI

클라이언트 VPN 엔드포인트의 경로를 삭제하려면

다음 delete-client-vpn-route 예제에서는 클라이언트 VPN 엔드포인트의 지정된 서브넷에 대한 0.0.0.0/0 경로를 삭제합니다.

```
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
```

출력:

```
{
    "Status": {
        "Code": "deleting"
    }
}
```

자세한 내용은 AWS 클라이언트 VPN 관리자 안내서의 라우팅을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteClientVpnRoute의 섹션을 참조하세요. AWS CLI

### delete-coip-cidr

다음 코드 예시에서는 delete-coip-cidr을 사용하는 방법을 보여 줍니다.

AWS CLI

고객 소유 IP(CoIP) 주소 범위를 삭제하려면

다음 delete-coip-cidr 예제에서는 지정된 CoIP 풀에서 지정된 범위의 CoIP 주소를 삭제합니다.

```
aws ec2 delete-coip-cidr \
    --cidr 14.0.0.0/24 \
    --coip-pool-id ipv4pool-coip-1234567890abcdefg
```

출력:

```
{
    "CoipCidr": {
        "Cidr": "14.0.0.0/24",
        "CoipPoolId": "ipv4pool-coip-1234567890abcdefg",
        "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890"
    }
}
```

자세한 내용은 AWS Outposts 사용 설명서의 고객 소유 IP 주소를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteCoipCidr의 섹션을 참조하세요. AWS CLI

### delete-coip-pool

다음 코드 예시에서는 delete-coip-pool을 사용하는 방법을 보여 줍니다.

AWS CLI

고객 소유 IP(CoIP) 주소 풀을 삭제하려면

다음 delete-coip-pool 예제에서는 CoIP CoIP 주소 풀을 삭제합니다.

```
aws ec2 delete-coip-pool \
    --coip-pool-id ipv4pool-coip-1234567890abcdefg
```

출력:

```
{
    "CoipPool": {
        "PoolId": "ipv4pool-coip-1234567890abcdefg",
        "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890",
```

```
        "PoolArn": "arn:aws:ec2:us-west-2:123456789012:coip-pool/ipv4pool-
coip-1234567890abcdefg"
    }
}
```

자세한 내용은 AWS Outposts 사용 설명서의 고객 소유 IP 주소를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteCoipPool의 섹션을 참조하세요. AWS CLI

### `delete-customer-gateway`

다음 코드 예시에서는 `delete-customer-gateway`을 사용하는 방법을 보여 줍니다.

AWS CLI

고객 게이트웨이를 삭제하려면

이 예제에서는 지정된 고객 게이트웨이를 삭제합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 delete-customer-gateway --customer-gateway-id cgw-0e11f167
```

- 자세한 API 내용은 명령 참조DeleteCustomerGateway의 섹션을 참조하세요. AWS CLI

### `delete-dhcp-options`

다음 코드 예시에서는 `delete-dhcp-options`을 사용하는 방법을 보여 줍니다.

AWS CLI

DHCP 옵션 세트를 삭제하려면

이 예제에서는 지정된 DHCP 옵션 세트를 삭제합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 delete-dhcp-options --dhcp-options-id dopt-d9070ebb
```

- 자세한 API 내용은 명령 참조DeleteDhcpOptions의 섹션을 참조하세요. AWS CLI

### `delete-egress-only-internet-gateway`

다음 코드 예시에서는 `delete-egress-only-internet-gateway`을 사용하는 방법을 보여 줍니다.

AWS CLI

송신 전용 인터넷 게이트웨이를 삭제하려면

이 예제에서는 지정된 송신 전용 인터넷 게이트웨이를 삭제합니다.

명령:

```
aws ec2 delete-egress-only-internet-gateway --egress-only-internet-gateway-
id eigw-01eadbd45ecd7943f
```

출력:

```
{
  "ReturnCode": true
}
```

- 자세한 API 내용은 명령 참조DeleteEgressOnlyInternetGateway의 섹션을 참조하세요. AWS CLI

### `delete-fleets`

다음 코드 예시에서는 `delete-fleets`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: EC2 플릿을 삭제하고 연결된 인스턴스를 종료하려면

다음 `delete-fleets` 예제에서는 지정된 EC2 플릿을 삭제하고 연결된 온디맨드 인스턴스 및 스팟 인스턴스를 종료합니다.

```
aws ec2 delete-fleets \
    --fleet-ids fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE \
    --terminate-instances
```

출력:

```
{
    "SuccessfulFleetDeletions": [
        {
            "CurrentFleetState": "deleted_terminating",
            "PreviousFleetState": "active",
            "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
        }
    ],
    "UnsuccessfulFleetDeletions": []
}
```

자세한 내용은 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 EC2 플릿 삭제를 참조하세요.

예제 2: 연결된 인스턴스를 종료하지 않고 EC2 플릿을 삭제하려면

다음 `delete-fleets` 예제에서는 연결된 온디맨드 인스턴스 및 스팟 인스턴스를 종료하지 않고 지정된 EC2 플릿을 삭제합니다.

```
aws ec2 delete-fleets \
    --fleet-ids fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE \
    --no-terminate-instances
```

출력:

```
{
    "SuccessfulFleetDeletions": [
        {
            "CurrentFleetState": "deleted_running",
            "PreviousFleetState": "active",
            "FleetId": "fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE"
        }
    ],
    "UnsuccessfulFleetDeletions": []
}
```

자세한 내용은 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 EC2 플릿 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteFleets의 섹션을 참조하세요. AWS CLI

### delete-flow-logs

다음 코드 예시에서는 delete-flow-logs을 사용하는 방법을 보여 줍니다.

AWS CLI

흐름 로그를 삭제하려면

다음 delete-flow-logs 예제에서는 지정된 흐름 로그를 삭제합니다.

```
aws ec2 delete-flow-logs --flow-log-id fl-11223344556677889
```

출력:

```
{
    "Unsuccessful": []
}
```

- 자세한 API 내용은 명령 참조DeleteFlowLogs의 섹션을 참조하세요. AWS CLI

### delete-fpga-image

다음 코드 예시에서는 delete-fpga-image을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon FPGA 이미지를 삭제하려면

이 예제에서는 지정된 를 삭제합니다AFI.

명령:

```
aws ec2 delete-fpga-image --fpga-image-id afi-06b12350a123fbabc
```

출력:

```
{
  "Return": true
}
```

- 자세한 API 내용은 명령 참조DeleteFpgaImage의 섹션을 참조하세요. AWS CLI

### delete-instance-connect-endpoint

다음 코드 예시에서는 delete-instance-connect-endpoint을 사용하는 방법을 보여 줍니다.

AWS CLI

EC2 인스턴스 연결 엔드포인트를 삭제하려면

다음 delete-instance-connect-endpoint 예제에서는 지정된 EC2 Instance Connect 엔드포인트를 삭제합니다.

```
aws ec2 delete-instance-connect-endpoint \
    --instance-connect-endpoint-id eice-03f5e49b83924bbc7
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon EC2 사용 설명서의 EC2 인스턴스 연결 엔드포인트 제거를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteInstanceConnectEndpoint의 섹션을 참조하세요. AWS CLI

### delete-instance-event-window

다음 코드 예시에서는 delete-instance-event-window을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 이벤트 창을 삭제하려면

다음 delete-instance-event-window 예제에서는 이벤트 창을 삭제합니다.

```
aws ec2 delete-instance-event-window \
    --region us-east-1 \
    --instance-event-window-id iew-0abcdef1234567890
```

출력:

```
{
    "InstanceEventWindowState": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "State": "deleting"
    }
}
```

이벤트 기간 제약 조건은 Amazon EC2 사용 설명서의 예약된 이벤트 섹션의 고려 사항을 참조하세요.

예제 2: 이벤트 창을 강제 삭제하려면

다음 delete-instance-event-window 예제에서는 이벤트 기간이 현재 대상과 연결되어 있는 경우 이벤트 창을 강제 삭제합니다.

```
aws ec2 delete-instance-event-window \
    --region us-east-1 \
    --instance-event-window-id iew-0abcdef1234567890 \
    --force-delete
```

출력:

```
{
    "InstanceEventWindowState": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "State": "deleting"
    }
}
```

이벤트 기간 제약 조건은 Amazon EC2 사용 설명서의 예약된 이벤트 섹션의 고려 사항을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteInstanceEventWindow의 섹션을 참조하세요. AWS CLI

### `delete-internet-gateway`

다음 코드 예시에서는 `delete-internet-gateway`을 사용하는 방법을 보여 줍니다.

AWS CLI

인터넷 게이트웨이를 삭제하려면

다음 `delete-internet-gateway` 예제에서는 지정된 인터넷 게이트웨이를 삭제합니다.

```
aws ec2 delete-internet-gateway \
    --internet-gateway-id igw-0d0fb496b3EXAMPLE
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon VPC 사용 설명서의 인터넷 게이트웨이를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteInternetGateway의 섹션을 참조하세요. AWS CLI

### `delete-ipam-pool`

다음 코드 예시에서는 `delete-ipam-pool`을 사용하는 방법을 보여 줍니다.

AWS CLI

IPAM 풀을 삭제하려면

이 예제에서는 더 이상 필요하지 않은 IPAM 풀을 삭제하려는 IPAM 위임된 관리자이지만 풀에 CIDR 프로비저닝된 가 있습니다. `--cascade` 옵션을 사용하지 않는 한 풀이 CIDRs 프로비저닝된 경우 풀을 삭제할 수 없으므로 를 사용합니다`--cascade`.

이 요청을 완료하려면:

로 가져올 수 있는 IPAM 풀 ID가 필요합니다describe-ipam-pools. 는 IPAM 홈 리전이어야 `--region` 합니다.

다음 delete-ipam-pool 예제에서는 AWS 계정의 IPAM 풀을 삭제합니다.

```
aws ec2 delete-ipam-pool \
    --ipam-pool-id ipam-pool-050c886a3ca41cd5b \
    --cascade \
    --region us-east-1
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 Amazon VPC IPAM 사용 설명서의 풀 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteIpamPool의 섹션을 참조하세요. AWS CLI

### delete-ipam-resource-discovery

다음 코드 예시에서는 delete-ipam-resource-discovery을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 검색을 삭제하려면

이 예제에서는 조직 외부의 IPAM 계정과 통합하는 과정에서 다른 관리자와 공유하기 위해 생성한 기본이 아닌 리소스 검색을 삭제하려는 IPAM 위임된 IPAM 관리자입니다.

이 요청을 완료하려면:

는 리소스 검색을 생성한 리전이어야 `--region` 합니다. 의 경우 기본 리소스 검색을 삭제할 수 없습니다`"IsDefault": true`. 기본 리소스 검색은 를 생성하는 계정에서 자동으로 생성되는 검색입니다IPAM. 기본 리소스 검색을 삭제하려면 를 삭제해야 합니다IPAM.

다음 `delete-ipam-resource-discovery` 예제에서는 리소스 검색을 삭제합니다.

```
aws ec2 delete-ipam-resource-discovery \
    --ipam-resource-discovery-id ipam-res-disco-0e39761475298ee0f \
    --region us-east-1
```

출력:

```
{
    "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"
    }
}
```

리소스 검색에 대한 자세한 내용은 Amazon VPC IPAM 사용 설명서의 리소스 검색 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteIpamResourceDiscovery의 섹션을 참조하세요. AWS CLI

### `delete-ipam-scope`

다음 코드 예시에서는 `delete-ipam-scope`을 사용하는 방법을 보여 줍니다.

AWS CLI

IPAM 범위를 삭제하려면

다음 delete-ipam-scope 예제에서는 를 삭제합니다IPAM.

```
aws ec2 delete-ipam-scope \
    --ipam-scope-id ipam-scope-01c1ebab2b63bd7e4
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon VPC IPAM 사용 설명서의 범위 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteIpamScope의 섹션을 참조하세요. AWS CLI

### delete-ipam

다음 코드 예시에서는 delete-ipam을 사용하는 방법을 보여 줍니다.

AWS CLI

를 삭제하려면 IPAM

다음 delete-ipam 예제에서는 를 삭제합니다IPAM.

```
aws ec2 delete-ipam \
```

```
    --ipam-id ipam-036486dfa6af58ee0
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon VPC IPAM 사용 설명서의 삭제IPAM를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteIpam의 섹션을 참조하세요. AWS CLI

### delete-key-pair

다음 코드 예시에서는 delete-key-pair을 사용하는 방법을 보여 줍니다.

AWS CLI

키 페어를 삭제하는 방법

다음 delete-key-pair 예제에서는 지정된 키 페어를 삭제합니다.

```
aws ec2 delete-key-pair \
```

```
    --key-name my-key-pair
```

출력:

```
{
    "Return": true,
    "KeyPairId": "key-03c8d3aceb53b507"
}
```

자세한 내용은 AWS 명령줄 인터페이스 사용 설명서의 키 페어 생성 및 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteKeyPair의 섹션을 참조하세요. AWS CLI

### delete-launch-template-versions

다음 코드 예시에서는 delete-launch-template-versions을 사용하는 방법을 보여 줍니다.

AWS CLI

시작 템플릿 버전을 삭제하려면

이 예제에서는 지정된 시작 템플릿 버전을 삭제합니다.

명령:

```
aws ec2 delete-launch-template-versions --launch-template-id lt-0abcd290751193123 --
versions 1
```

출력:

```
{
  "UnsuccessfullyDeletedLaunchTemplateVersions": [],
  "SuccessfullyDeletedLaunchTemplateVersions": [
      {
          "LaunchTemplateName": "TestVersion",
          "VersionNumber": 1,
          "LaunchTemplateId": "lt-0abcd290751193123"
      }
  ]
}
```

- 자세한 API 내용은 명령 참조DeleteLaunchTemplateVersions의 섹션을 참조하세요. AWS CLI

### delete-launch-template

다음 코드 예시에서는 delete-launch-template을 사용하는 방법을 보여 줍니다.

AWS CLI

시작 템플릿을 삭제하는 방법

다음 예제에서는 지정된 시작 템플릿을 삭제합니다.

명령:

```
aws ec2 delete-launch-template --launch-template-id lt-0abcd290751193123
```

출력:

```
{
  "LaunchTemplate": {
      "LatestVersionNumber": 2,
      "LaunchTemplateId": "lt-0abcd290751193123",
      "LaunchTemplateName": "TestTemplate",
      "DefaultVersionNumber": 2,
      "CreatedBy": "arn:aws:iam::123456789012:root",
      "CreateTime": "2017-11-23T16:46:25.000Z"
  }
}
```

- 자세한 API 내용은 명령 참조DeleteLaunchTemplate의 섹션을 참조하세요. AWS CLI

### delete-local-gateway-route-table-virtual-interface-group-association

다음 코드 예시에서는 delete-local-gateway-route-table-virtual-interface-group-association을 사용하는 방법을 보여 줍니다.

AWS CLI

로컬 게이트웨이 라우팅 테이블을 가상 인터페이스(VIFs) 그룹에서 연결 해제하려면

다음 delete-local-gateway-route-table-virtual-interface-group-association 예제에서는 지정된 로컬 게이트웨이 라우팅 테이블과 VIF 그룹 간의 연결을 삭제합니다.

```
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
```

출력:

```
{
    "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": []
    }
}
```

자세한 내용은 Outposts 사용 설명서의 VIF 그룹 연결을 참조하세요. AWS

- 자세한 API 내용은 명령 참조DeleteLocalGatewayRouteTableVirtualInterfaceGroupAssociation의 섹션을 참조하세요. AWS CLI

### delete-local-gateway-route-table-vpc-association

다음 코드 예시에서는 delete-local-gateway-route-table-vpc-association을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 로컬 게이트웨이 라우팅 테이블의 연결을 해제하려면 VPC

다음 delete-local-gateway-route-table-vpc-association 예제에서는 지정된 로컬 게이트웨이 라우팅 테이블과 간의 연결을 삭제합니다VPC.

```
aws ec2 delete-local-gateway-route-table-vpc-association \
    --local-gateway-route-table-vpc-association-id vpc-example0123456789
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Outposts 사용 설명서의 VPC 연결을 참조하세요. AWS

- 자세한 API 내용은 명령 참조DeleteLocalGatewayRouteTableVpcAssociation의 섹션을 참조하세요. AWS CLI

### `delete-local-gateway-route-table`

다음 코드 예시에서는 `delete-local-gateway-route-table`을 사용하는 방법을 보여 줍니다.

AWS CLI

로컬 게이트웨이 라우팅 테이블을 삭제하려면

다음 `delete-local-gateway-route-table` 예제에서는 직접 라우팅 모드를 사용하여 로컬 게이트웨이 VPC 라우팅 테이블을 생성합니다.

```
aws ec2 delete-local-gateway-route-table \
    --local-gateway-route-table-id lgw-rtb-abcdefg1234567890
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS Outposts 사용 설명서의 로컬 게이트웨이 라우팅 테이블을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteLocalGatewayRouteTable의 섹션을 참조하세요. AWS CLI

### `delete-local-gateway-route`

다음 코드 예시에서는 `delete-local-gateway-route`을 사용하는 방법을 보여 줍니다.

AWS CLI

로컬 게이트웨이 라우팅 테이블에서 라우팅을 삭제하려면

다음 `delete-local-gateway-route` 예제에서는 지정된 로컬 게이트웨이 라우팅 테이블에서 지정된 라우팅을 삭제합니다.

```
aws ec2 delete-local-gateway-route \
    --destination-cidr-block 0.0.0.0/0 \
    --local-gateway-route-table-id lgw-rtb-059615ef7dEXAMPLE
```

출력:

```
{
    "Route": {
        "DestinationCidrBlock": "0.0.0.0/0",
        "LocalGatewayVirtualInterfaceGroupId": "lgw-vif-grp-07145b276bEXAMPLE",
        "Type": "static",
        "State": "deleted",
        "LocalGatewayRouteTableId": "lgw-rtb-059615ef7EXAMPLE"
    }
}
```

- 자세한 API 내용은 명령 참조DeleteLocalGatewayRoute의 섹션을 참조하세요. AWS CLI

### delete-managed-prefix-list

다음 코드 예시에서는 delete-managed-prefix-list을 사용하는 방법을 보여 줍니다.

AWS CLI

접두사 목록을 삭제하려면

다음 delete-managed-prefix-list 예제에서는 지정된 접두사 목록을 삭제합니다.

```
aws ec2 delete-managed-prefix-list \
    --prefix-list-id pl-0123456abcabcabc1
```

출력:

```
{
    "PrefixList": {
        "PrefixListId": "pl-0123456abcabcabc1",
        "AddressFamily": "IPv4",
        "State": "delete-in-progress",
        "PrefixListArn": "arn:aws:ec2:us-west-2:123456789012:prefix-list/
pl-0123456abcabcabc1",
        "PrefixListName": "test",
        "MaxEntries": 10,
        "Version": 1,
        "OwnerId": "123456789012"
    }
}
```

자세한 내용은 Amazon VPC 사용 설명서의 관리형 접두사 목록을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteManagedPrefixList의 섹션을 참조하세요. AWS CLI

### delete-nat-gateway

다음 코드 예시에서는 delete-nat-gateway을 사용하는 방법을 보여 줍니다.

AWS CLI

NAT 게이트웨이를 삭제하려면

이 예제에서는 NAT 게이트웨이 를 삭제합니다nat-04ae55e711cec5680.

명령:

```
aws ec2 delete-nat-gateway --nat-gateway-id nat-04ae55e711cec5680
```

출력:

```
{
    "NatGatewayId": "nat-04ae55e711cec5680"
}
```

- 자세한 API 내용은 명령 참조DeleteNatGateway의 섹션을 참조하세요. AWS CLI

### delete-network-acl-entry

다음 코드 예시에서는 delete-network-acl-entry을 사용하는 방법을 보여 줍니다.

AWS CLI

네트워크 ACL 항목을 삭제하려면

이 예제에서는 지정된 네트워크 에서 수신 규칙 번호 100을 삭제합니다ACL. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 delete-network-acl-entry --network-acl-id acl-5fb85d36 --ingress --rule-number 100
```

- 자세한 API 내용은 명령 참조DeleteNetworkAclEntry의 섹션을 참조하세요. AWS CLI

### delete-network-acl

다음 코드 예시에서는 delete-network-acl을 사용하는 방법을 보여 줍니다.

AWS CLI

네트워크를 삭제하려면 ACL

이 예제에서는 지정된 네트워크 를 삭제합니다ACL. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 delete-network-acl --network-acl-id acl-5fb85d36
```

- 자세한 API 내용은 명령 참조DeleteNetworkAcl의 섹션을 참조하세요. AWS CLI

### delete-network-insights-access-scope-analysis

다음 코드 예시에서는 delete-network-insights-access-scope-analysis을 사용하는 방법을 보여 줍니다.

AWS CLI

네트워크 액세스 범위 분석을 삭제하려면

다음 delete-network-insights-access-scope-analysis 예제에서는 지정된 네트워크 액세스 범위 분석을 삭제합니다.

```
aws ec2 delete-network-insights-access-scope-analysis \
    --network-insights-access-scope-analysis-id nisa-01234567891abcdef
```

출력:

```
{
    "NetworkInsightsAccessScopeAnalysisId": "nisa-01234567891abcdef
}
```

자세한 내용은 Network Access Analyzer 가이드의 를 사용하여 AWS CLI Network Access Analyzer 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteNetworkInsightsAccessScopeAnalysis의 섹션을 참조하세요. AWS CLI

### delete-network-insights-access-scope

다음 코드 예시에서는 delete-network-insights-access-scope을 사용하는 방법을 보여 줍니다.

AWS CLI

네트워크 액세스 범위를 삭제하려면

다음 delete-network-insights-access-scope 예제에서는 지정된 네트워크 액세스 범위를 삭제합니다.

```
aws ec2 delete-network-insights-access-scope \
    --network-insights-access-scope-id nis-123456789abc01234
```

출력:

```
{
    "NetworkInsightsAccessScopeId": "nis-123456789abc01234"
}
```

자세한 내용은 Network Access Analyzer 가이드의 를 사용하여 AWS CLI Network Access Analyzer 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteNetworkInsightsAccessScope의 섹션을 참조하세요. AWS CLI

### delete-network-insights-analysis

다음 코드 예시에서는 delete-network-insights-analysis을 사용하는 방법을 보여 줍니다.

AWS CLI

경로 분석을 삭제하려면

다음 delete-network-insights-analysis 예제에서는 지정된 분석을 삭제합니다.

```
aws ec2 delete-network-insights-analysis \
    --network-insights-analysis-id nia-02207aa13eb480c7a
```

출력:

```
{
    "NetworkInsightsAnalysisId": "nia-02207aa13eb480c7a"
}
```

자세한 내용은 Reachability Analyzer 안내서의 시작하기 AWS CLI를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteNetworkInsightsAnalysis의 섹션을 참조하세요. AWS CLI

### delete-network-insights-path

다음 코드 예시에서는 delete-network-insights-path을 사용하는 방법을 보여 줍니다.

AWS CLI

경로를 삭제하려면

다음 delete-network-insights-path 예제에서는 지정된 경로를 삭제합니다. 경로를 삭제하려면 먼저 delete-network-insights-analysis 명령을 사용하여 모든 분석을 삭제해야 합니다.

```
aws ec2 delete-network-insights-path \
    --network-insights-path-id nip-0b26f224f1d131fa8
```

출력:

```
{
    "NetworkInsightsPathId": "nip-0b26f224f1d131fa8"
}
```

자세한 내용은 Reachability Analyzer 가이드의 시작하기 AWS CLI를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteNetworkInsightsPath의 섹션을 참조하세요. AWS CLI

### delete-network-interface-permission

다음 코드 예시에서는 delete-network-interface-permission을 사용하는 방법을 보여 줍니다.

AWS CLI

네트워크 인터페이스 권한을 삭제하려면

이 예제에서는 지정된 네트워크 인터페이스 권한을 삭제합니다.

명령:

```
aws ec2 delete-network-interface-permission --network-interface-permission-id eni-perm-06fd19020ede149ea
```

출력:

```
{
  "Return": true
}
```

- 자세한 API 내용은 명령 참조DeleteNetworkInterfacePermission의 섹션을 참조하세요. AWS CLI

### `delete-network-interface`

다음 코드 예시에서는 `delete-network-interface`을 사용하는 방법을 보여 줍니다.

AWS CLI

네트워크 인터페이스를 삭제하려면

이 예제에서는 지정된 네트워크 인터페이스를 삭제합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 delete-network-interface --network-interface-id eni-e5aa89a3
```

- 자세한 API 내용은 명령 참조DeleteNetworkInterface의 섹션을 참조하세요. AWS CLI

### `delete-placement-group`

다음 코드 예시에서는 `delete-placement-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

배치 그룹을 삭제하려면

이 예제 명령은 지정된 배치 그룹을 삭제합니다.

명령:

```
aws ec2 delete-placement-group --group-name my-cluster
```

- 자세한 API 내용은 명령 참조DeletePlacementGroup의 섹션을 참조하세요. AWS CLI

### delete-queued-reserved-instances

다음 코드 예시에서는 delete-queued-reserved-instances을 사용하는 방법을 보여 줍니다.

AWS CLI

대기열에 있는 구매를 삭제하려면

다음 delete-queued-reserved-instances 예제에서는 구매를 위해 대기열에 있는 지정된 예약 인스턴스를 삭제합니다.

```
aws ec2 delete-queued-reserved-instances \
    --reserved-instances-ids af9f760e-6f91-4559-85f7-4980eexample
```

출력:

```
{
    "SuccessfulQueuedPurchaseDeletions": [
        {
            "ReservedInstancesId": "af9f760e-6f91-4559-85f7-4980eexample"
        }
    ],
    "FailedQueuedPurchaseDeletions": []
}
```

- 자세한 API 내용은 명령 참조DeleteQueuedReservedInstances의 섹션을 참조하세요. AWS CLI

### delete-route-table

다음 코드 예시에서는 delete-route-table을 사용하는 방법을 보여 줍니다.

AWS CLI

라우팅 테이블을 삭제하려면

이 예제에서는 지정된 라우팅 테이블을 삭제합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 delete-route-table --route-table-id rtb-22574640
```

- 자세한 API 내용은 명령 참조DeleteRouteTable의 섹션을 참조하세요. AWS CLI

### delete-route

다음 코드 예시에서는 delete-route을 사용하는 방법을 보여 줍니다.

AWS CLI

라우팅을 삭제하려면

이 예제에서는 지정된 라우팅 테이블에서 지정된 라우팅을 삭제합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 delete-route --route-table-id rtb-22574640 --destination-cidr-
block 0.0.0.0/0
```

- 자세한 API 내용은 명령 참조DeleteRoute의 섹션을 참조하세요. AWS CLI

### delete-security-group

다음 코드 예시에서는 delete-security-group을 사용하는 방법을 보여 줍니다.

AWS CLI

[EC2-Classic] 보안 그룹을 삭제하려면

이 예제에서는 이름이 MySecurityGroup인 보안 그룹을 삭제합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 delete-security-group --group-name MySecurityGroup
```

[EC2-VPC] 보안 그룹을 삭제하려면

이 예제에서는 ID가 sg-903004f8인 보안 그룹을 삭제합니다. 에 대한VPC 보안 그룹을 EC2이름으로 참조할 수 없습니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 delete-security-group --group-id sg-903004f8
```

자세한 내용은 AWS Command Line Interface 사용 설명서의 보안 그룹 사용을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteSecurityGroup의 섹션을 참조하세요. AWS CLI

### delete-snapshot

다음 코드 예시에서는 delete-snapshot을 사용하는 방법을 보여 줍니다.

AWS CLI

스냅샷을 삭제하는 방법

이 예제 명령은 스냅샷 ID가 snap-1234567890abcdef0인 스냅샷을 삭제합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 delete-snapshot --snapshot-id snap-1234567890abcdef0
```

- 자세한 API 내용은 명령 참조DeleteSnapshot의 섹션을 참조하세요. AWS CLI

### delete-spot-datafeed-subscription

다음 코드 예시에서는 delete-spot-datafeed-subscription을 사용하는 방법을 보여 줍니다.

AWS CLI

스팟 인스턴스 데이터 피드 구독을 취소하려면

이 예제 명령은 계정에 대한 스팟 데이터 피드 구독을 삭제합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 delete-spot-datafeed-subscription
```

- 자세한 API 내용은 명령 참조DeleteSpotDatafeedSubscription의 섹션을 참조하세요. AWS CLI

### delete-subnet-cidr-reservation

다음 코드 예시에서는 delete-subnet-cidr-reservation을 사용하는 방법을 보여 줍니다.

AWS CLI

서브넷 CIDR 예약을 삭제하려면

다음 delete-subnet-cidr-reservation 예제에서는 지정된 서브넷 CIDR 예약을 삭제합니다.

```
aws ec2 delete-subnet-cidr-reservation \
    --subnet-cidr-reservation-id scr-044f977c4eEXAMPLE
```

출력:

```
{
    "DeletedSubnetCidrReservation": {
        "SubnetCidrReservationId": "scr-044f977c4eEXAMPLE",
        "SubnetId": "subnet-03c51e2e6cEXAMPLE",
        "Cidr": "10.1.0.16/28",
        "ReservationType": "prefix",
        "OwnerId": "123456789012"
    }
}
```

자세한 내용은 Amazon VPC 사용 설명서의 서브넷 CIDR 예약을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteSubnetCidrReservation의 섹션을 참조하세요. AWS CLI

### delete-subnet

다음 코드 예시에서는 delete-subnet을 사용하는 방법을 보여 줍니다.

AWS CLI

서브넷을 삭제하려면

이 예제에서는 지정된 서브넷을 삭제합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 delete-subnet --subnet-id subnet-9d4a7b6c
```

- 자세한 API 내용은 명령 참조DeleteSubnet의 섹션을 참조하세요. AWS CLI

### delete-tags

다음 코드 예시에서는 delete-tags을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 리소스에서 태그를 삭제하려면

다음 delete-tags 예제에서는 지정된 이미지Stack=Test에서 태그를 삭제합니다. 값과 키 이름을 모두 지정하면 태그의 값이 지정된 값과 일치하는 경우에만 태그가 삭제됩니다.

```
aws ec2 delete-tags \
    --resources ami-1234567890abcdef0 \
    --tags Key=Stack,Value=Test
```

태그 값을 지정하는 것은 선택 사항입니다. 다음 delete-tags 예제에서는 태그의 태그 값에 관계없이 지정된 인스턴스purpose에서 키 이름이 인 태그를 삭제합니다.

```
aws ec2 delete-tags \
    --resources i-1234567890abcdef0 \
    --tags Key=purpose
```

빈 문자열을 태그 값으로 지정하면 태그의 값이 빈 문자열인 경우에만 태그가 삭제됩니다. 다음 delete-tags 예제에서는 빈 문자열을 삭제할 태그의 태그 값으로 지정합니다.

```
aws ec2 delete-tags \
    --resources i-1234567890abcdef0 \
    --tags Key=Name,Value=
```

예제 2: 여러 리소스에서 태그를 삭제하려면

다음 delete-tags 예제에서는 인스턴스와 에서 모두 태그``Purpose=Test``를 삭제합니다AMI. 이전 예제와 같이 명령에서 태그 값을 생략할 수 있습니다.

```
aws ec2 delete-tags \
    --resources i-1234567890abcdef0 ami-1234567890abcdef0 \
    --tags Key=Purpose
```

- 자세한 API 내용은 명령 참조DeleteTags의 섹션을 참조하세요. AWS CLI

### delete-traffic-mirror-filter-rule

다음 코드 예시에서는 delete-traffic-mirror-filter-rule을 사용하는 방법을 보여 줍니다.

AWS CLI

트래픽 미러 필터 규칙을 삭제하려면

다음 delete-traffic-mirror-filter-rule 예제에서는 지정된 트래픽 미러 필터 규칙을 삭제합니다.

```
aws ec2 delete-traffic-mirror-filter-rule \
    --traffic-mirror-filter-rule-id tmfr-081f71283bEXAMPLE
```

출력:

```
{
    "TrafficMirrorFilterRuleId": "tmfr-081f71283bEXAMPLE"
}
```

자세한 내용은 트래픽 미러링 가이드의 트래픽 미러 필터 규칙 수정을 참조하세요. AWS

- 자세한 API 내용은 명령 참조DeleteTrafficMirrorFilterRule의 섹션을 참조하세요. AWS CLI

### delete-traffic-mirror-filter

다음 코드 예시에서는 delete-traffic-mirror-filter을 사용하는 방법을 보여 줍니다.

AWS CLI

트래픽 미러 필터를 삭제하려면

다음 delete-traffic-mirror-filter 예제에서는 지정된 트래픽 미러 필터를 삭제합니다.

```
aws ec2 delete-traffic-mirror-filter \
    --traffic-mirror-filter-id tmf-0be0b25fcdEXAMPLE
```

출력:

```
{
    "TrafficMirrorFilterId": "tmf-0be0b25fcdEXAMPLE"
}
```

자세한 내용은 트래픽 미러링 가이드의 트래픽 미러 필터 삭제를 참조하세요. AWS

- 자세한 API 내용은 명령 참조DeleteTrafficMirrorFilter의 섹션을 참조하세요. AWS CLI

### delete-traffic-mirror-session

다음 코드 예시에서는 delete-traffic-mirror-session을 사용하는 방법을 보여 줍니다.

AWS CLI

트래픽 미러 세션을 삭제하려면

다음 delete-traffic-mirror-session 예제에서는 지정된 트래픽 미러 세션을 삭제합니다.

```
aws ec2 delete-traffic-mirror-session \
    --traffic-mirror-session-id tms-0af3141ce5EXAMPLE
```

출력:

```
{
    "TrafficMirrorSessionId": "tms-0af3141ce5EXAMPLE"
}
```

자세한 내용은 트래픽 미러링 가이드의 트래픽 미러 세션 삭제를 참조하세요. AWS

- 자세한 API 내용은 명령 참조DeleteTrafficMirrorSession의 섹션을 참조하세요. AWS CLI

### delete-traffic-mirror-target

다음 코드 예시에서는 delete-traffic-mirror-target을 사용하는 방법을 보여 줍니다.

AWS CLI

트래픽 미러 대상을 삭제하려면

다음 delete-traffic-mirror-target 예제에서는 지정된 트래픽 미러 대상을 삭제합니다.

```
aws ec2 delete-traffic-mirror-target \
    --traffic-mirror-target-id tmt-060f48ce9EXAMPLE
```

출력:

```
{
    "TrafficMirrorTargetId": "tmt-060f48ce9EXAMPLE"
}
```

자세한 내용은 트래픽 미러링 가이드의 트래픽 미러 대상 삭제를 참조하세요. AWS

- 자세한 API 내용은 명령 참조DeleteTrafficMirrorTarget의 섹션을 참조하세요. AWS CLI

### delete-transit-gateway-connect-peer

다음 코드 예시에서는 delete-transit-gateway-connect-peer을 사용하는 방법을 보여 줍니다.

AWS CLI

Transit Gateway Connect 피어를 삭제하려면

다음 delete-transit-gateway-connect-peer 예제에서는 지정된 Connect 피어를 삭제합니다.

```
aws ec2 delete-transit-gateway-connect-peer \
    --transit-gateway-connect-peer-id tgw-connect-peer-0666adbac4EXAMPLE
```

출력:

```
{
    "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"
                }
            ]
        }
    }
}
```

자세한 내용은 Transit Gateways 가이드의 Transit Gateway Connect 연결 및 Transit Gateway Connect 피어를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteTransitGatewayConnectPeer의 섹션을 참조하세요. AWS CLI

### `delete-transit-gateway-connect`

다음 코드 예시에서는 `delete-transit-gateway-connect`을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 Connect 연결 삭제

다음 delete-transit-gateway-connect 예제에서는 지정된 Connect 연결을 삭제합니다.

```
aws ec2 delete-transit-gateway-connect \
    --transit-gateway-attachment-id tgw-attach-037012e5dcEXAMPLE
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 Transit Gateways 가이드의 Transit Gateway Connect 연결 및 Transit Gateway Connect 피어를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteTransitGatewayConnect의 섹션을 참조하세요. AWS CLI

### delete-transit-gateway-multicast-domain

다음 코드 예시에서는 delete-transit-gateway-multicast-domain을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 멀티캐스트 도메인을 삭제하려면

다음 delete-transit-gateway-multicast-domain 예제에서는 지정된 멀티캐스트 도메인을 삭제합니다.

```
aws ec2 delete-transit-gateway-multicast-domain \
    --transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef7EXAMPLE
```

출력:

```
{
    "TransitGatewayMulticastDomain": {
        "TransitGatewayMulticastDomainId": "tgw-mcast-domain-02bb79002bEXAMPLE",
        "TransitGatewayId": "tgw-0d88d2d0d5EXAMPLE",
        "State": "deleting",
        "CreationTime": "2019-11-20T22:02:03.000Z"
    }
}
```

자세한 내용은 Transit Gateways 가이드의 멀티캐스트 도메인 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteTransitGatewayMulticastDomain의 섹션을 참조하세요. AWS CLI

### `delete-transit-gateway-peering-attachment`

다음 코드 예시에서는 `delete-transit-gateway-peering-attachment`을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 피어링 연결 삭제

다음 `delete-transit-gateway-peering-attachment` 예제에서는 지정된 전송 게이트웨이 피어링 연결을 삭제합니다.

```
aws ec2 delete-transit-gateway-peering-attachment \
    --transit-gateway-attachment-id tgw-attach-4455667788aabbccd
```

출력:

```
{
    "TransitGatewayPeeringAttachment": {
        "TransitGatewayAttachmentId": "tgw-attach-4455667788aabbccd",
        "RequesterTgwInfo": {
            "TransitGatewayId": "tgw-123abc05e04123abc",
            "OwnerId": "123456789012",
            "Region": "us-west-2"
        },
        "AccepterTgwInfo": {
            "TransitGatewayId": "tgw-11223344aabbcc112",
```

```
            "OwnerId": "123456789012",
            "Region": "us-east-2"
        },
        "State": "deleting",
        "CreationTime": "2019-12-09T11:38:31.000Z"
    }
}
```

자세한 내용은 Transit Gateways 가이드의 Transit Gateway 피어링 연결을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteTransitGatewayPeeringAttachment의 섹션을 참조하세요. AWS CLI

### `delete-transit-gateway-policy-table`

다음 코드 예시에서는 `delete-transit-gateway-policy-table`을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 정책 테이블을 삭제하려면

다음 `delete-transit-gateway-policy-table` 예제에서는 지정된 전송 게이트웨이 정책 테이블을 삭제합니다.

```
aws ec2 delete-transit-gateway-policy-table \
    --transit-gateway-policy-table-id tgw-ptb-0a16f134b78668a81
```

출력:

```
{
    "TransitGatewayPolicyTables": [
        {
            "TransitGatewayPolicyTableId": "tgw-ptb-0a16f134b78668a81",
            "TransitGatewayId": "tgw-067f8505c18f0bd6e",
            "State": "deleting",
            "CreationTime": "2023-11-28T16:36:43+00:00",
            "Tags": []
        }
    ]
}
```

자세한 내용은 Transit Gateway 사용 설명서의 Transit Gateway 정책 테이블을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteTransitGatewayPolicyTable의 섹션을 참조하세요. AWS CLI

### delete-transit-gateway-prefix-list-reference

다음 코드 예시에서는 delete-transit-gateway-prefix-list-reference을 사용하는 방법을 보여 줍니다.

AWS CLI

접두사 목록 참조를 삭제하려면

다음 delete-transit-gateway-prefix-list-reference 예제에서는 지정된 접두사 목록 참조를 삭제합니다.

```
aws ec2 delete-transit-gateway-prefix-list-reference \
    --transit-gateway-route-table-id tgw-rtb-0123456789abcd123 \
    --prefix-list-id pl-11111122222222333
```

출력:

```
{
    "TransitGatewayPrefixListReference": {
        "TransitGatewayRouteTableId": "tgw-rtb-0123456789abcd123",
        "PrefixListId": "pl-11111122222222333",
        "PrefixListOwnerId": "123456789012",
        "State": "deleting",
        "Blackhole": false,
        "TransitGatewayAttachment": {
            "TransitGatewayAttachmentId": "tgw-attach-aabbccddaabbccaab",
            "ResourceType": "vpc",
            "ResourceId": "vpc-112233445566aabbc"
        }
    }
}
```

자세한 내용은 Transit Gateways 가이드의 접두사 목록 참조를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteTransitGatewayPrefixListReference의 섹션을 참조하세요. AWS CLI

### delete-transit-gateway-route-table

다음 코드 예시에서는 delete-transit-gateway-route-table을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 라우팅 테이블을 삭제하려면

다음 delete-transit-gateway-route-table 예제에서는 지정된 전송 게이트웨이 라우팅 테이블을 삭제합니다.

```
aws ec2  delete-transit-gateway-route-table \
    --transit-gateway-route-table-id tgw-rtb-0b6f6aaa01EXAMPLE
```

출력:

```
{
    "TransitGatewayRouteTable": {
        "TransitGatewayRouteTableId": "tgw-rtb-0b6f6aaa01EXAMPLE",
        "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
        "State": "deleting",
        "DefaultAssociationRouteTable": false,
        "DefaultPropagationRouteTable": false,
        "CreationTime": "2019-07-17T20:27:26.000Z"
    }
}
```

자세한 내용은 Transit Gateways 가이드의 전송 게이트웨이 라우팅 테이블 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteTransitGatewayRouteTable의 섹션을 참조하세요. AWS CLI

### delete-transit-gateway-route

다음 코드 예시에서는 delete-transit-gateway-route을 사용하는 방법을 보여 줍니다.

AWS CLI

라우팅 테이블에서 CIDR 블록을 삭제하려면

다음 delete-transit-gateway-route 예제에서는 지정된 전송 게이트웨이 라우팅 테이블에서 CIDR 블록을 삭제합니다.

```
aws ec2 delete-transit-gateway-route \
    --transit-gateway-route-table-id tgw-rtb-0b6f6aaa01EXAMPLE \
    --destination-cidr-block 10.0.2.0/24
```

출력:

```
{
    "Route": {
        "DestinationCidrBlock": "10.0.2.0/24",
        "TransitGatewayAttachments": [
            {
                "ResourceId": "vpc-0065acced4EXAMPLE",
                "TransitGatewayAttachmentId": "tgw-attach-0b5968d3b6EXAMPLE",
                "ResourceType": "vpc"
            }
        ],
        "Type": "static",
        "State": "deleted"
    }
}
```

자세한 내용은 Transit Gateways 가이드의 정적 경로 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteTransitGatewayRoute의 섹션을 참조하세요. AWS CLI

### `delete-transit-gateway-vpc-attachment`

다음 코드 예시에서는 `delete-transit-gateway-vpc-attachment`을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 VPC 연결 삭제

다음 `delete-transit-gateway-vpc-attachment` 예제에서는 지정된 VPC 첨부 파일을 삭제합니다.

```
aws ec2 delete-transit-gateway-vpc-attachment \
    --transit-gateway-attachment-id tgw-attach-0d2c54bdbEXAMPLE
```

출력:

```
{
    "TransitGatewayVpcAttachment": {
        "TransitGatewayAttachmentId": "tgw-attach-0d2c54bdb3EXAMPLE",
        "TransitGatewayId": "tgw-02f776b1a7EXAMPLE",
        "VpcId": "vpc-0065acced4f61c651",
        "VpcOwnerId": "111122223333",
        "State": "deleting",
        "CreationTime": "2019-07-17T16:04:27.000Z"
    }
}
```

자세한 내용은 Transit Gateways 가이드의 VPC 첨부 파일 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteTransitGatewayVpcAttachment의 섹션을 참조하세요. AWS CLI

### delete-transit-gateway

다음 코드 예시에서는 delete-transit-gateway을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이를 삭제하려면

다음 delete-transit-gateway 예제에서는 지정된 전송 게이트웨이를 삭제합니다.

```
aws ec2  delete-transit-gateway \
    --transit-gateway-id tgw-01f04542b2EXAMPLE
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 Transit Gateways 가이드의 전송 게이트웨이 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteTransitGateway의 섹션을 참조하세요. AWS CLI

### `delete-verified-access-endpoint`

다음 코드 예시에서는 `delete-verified-access-endpoint`을 사용하는 방법을 보여 줍니다.

AWS CLI

Verified Access 엔드포인트를 삭제하려면

다음 `delete-verified-access-endpoint` 예제에서는 지정된 Verified Access 엔드포인트를 삭제합니다.

```
aws ec2 delete-verified-access-endpoint \
    --verified-access-endpoint-id vae-066fac616d4d546f2
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Verified Access 사용 설명서의 Verified Access 엔드포인트를 참조하세요. AWS

- 자세한 API 내용은 명령 참조DeleteVerifiedAccessEndpoint의 섹션을 참조하세요. AWS CLI

### delete-verified-access-group

다음 코드 예시에서는 delete-verified-access-group을 사용하는 방법을 보여 줍니다.

AWS CLI

Verified Access 그룹을 삭제하려면

다음 delete-verified-access-group 예제에서는 지정된 Verified Access 그룹을 삭제합니다.

```
aws ec2 delete-verified-access-group \
    --verified-access-group-id vagr-0dbe967baf14b7235
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Verified Access 사용 설명서의 Verified Access 그룹을 참조하세요. AWS

- 자세한 API 내용은 명령 참조DeleteVerifiedAccessGroup의 섹션을 참조하세요. AWS CLI

### `delete-verified-access-instance`

다음 코드 예시에서는 `delete-verified-access-instance`을 사용하는 방법을 보여 줍니다.

AWS CLI

Verified Access 인스턴스를 삭제하려면

다음 `delete-verified-access-instance` 예제에서는 지정된 Verified Access 인스턴스를 삭제합니다.

```
aws ec2 delete-verified-access-instance \
    --verified-access-instance-id vai-0ce000c0b7643abea
```

출력:

```
{
    "VerifiedAccessInstance": {
        "VerifiedAccessInstanceId": "vai-0ce000c0b7643abea",
        "Description": "Testing Verified Access",
        "VerifiedAccessTrustProviders": [],
        "CreationTime": "2023-08-25T18:27:56",
        "LastUpdatedTime": "2023-08-26T01:00:18"
    }
}
```

자세한 내용은 Verified Access 사용 설명서의 Verified Access 인스턴스를 참조하세요. AWS

- 자세한 API 내용은 명령 참조DeleteVerifiedAccessInstance의 섹션을 참조하세요. AWS CLI

### delete-verified-access-trust-provider

다음 코드 예시에서는 delete-verified-access-trust-provider을 사용하는 방법을 보여 줍니다.

AWS CLI

Verified Access 신뢰 공급자를 삭제하려면

다음 delete-verified-access-trust-provider 예제에서는 지정된 Verified Access 신뢰 공급자를 삭제합니다.

```
aws ec2 delete-verified-access-trust-provider \
    --verified-access-trust-provider-id vatp-0bb32de759a3e19e7
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Verified Access 사용 설명서의 Verified Access에 대한 신뢰 공급자를 참조하세요. AWS

- 자세한 API 내용은 명령 참조DeleteVerifiedAccessTrustProvider의 섹션을 참조하세요. AWS CLI

### delete-volume

다음 코드 예시에서는 delete-volume을 사용하는 방법을 보여 줍니다.

AWS CLI

볼륨을 삭제하려면

이 예제 명령은 볼륨 ID가 인 사용 가능한 볼륨을 삭제합니다vol-049df61146c4d7901. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 delete-volume --volume-id vol-049df61146c4d7901
```

- 자세한 API 내용은 명령 참조DeleteVolume의 섹션을 참조하세요. AWS CLI

### delete-vpc-endpoint-connection-notifications

다음 코드 예시에서는 delete-vpc-endpoint-connection-notifications을 사용하는 방법을 보여 줍니다.

AWS CLI

엔드포인트 연결 알림을 삭제하려면

이 예제에서는 지정된 엔드포인트 연결 알림을 삭제합니다.

명령:

```
aws ec2 delete-vpc-endpoint-connection-notifications --connection-notification-
ids vpce-nfn-008776de7e03f5abc
```

출력:

```
{
  "Unsuccessful": []
}
```

- 자세한 API 내용은 명령 참조DeleteVpcEndpointConnectionNotifications의 섹션을 참조하세요. AWS CLI

### delete-vpc-endpoint-service-configurations

다음 코드 예시에서는 delete-vpc-endpoint-service-configurations을 사용하는 방법을 보여 줍니다.

AWS CLI

엔드포인트 서비스 구성을 삭제하려면

이 예제에서는 지정된 엔드포인트 서비스 구성을 삭제합니다.

명령:

```
aws ec2 delete-vpc-endpoint-service-configurations --service-ids vpce-svc-03d5ebb7d9579a2b3
```

출력:

```
{
    "Unsuccessful": []
}
```

- 자세한 API 내용은 명령 참조DeleteVpcEndpointServiceConfigurations의 섹션을 참조하세요. AWS CLI

### `delete-vpc-endpoints`

다음 코드 예시에서는 `delete-vpc-endpoints`을 사용하는 방법을 보여 줍니다.

AWS CLI

엔드포인트를 삭제하려면

이 예제에서는 엔드포인트 vpce-aa22bb33 및 vpce-1a2b3c4d를 삭제합니다. 명령이 부분적으로 성공하거나 실패한 경우 실패한 항목 목록이 반환됩니다. 명령이 성공하면 반환된 목록이 비어 있습니다.

명령:

```
aws ec2 delete-vpc-endpoints --vpc-endpoint-ids vpce-aa22bb33 vpce-1a2b3c4d
```

출력:

```
{
  "Unsuccessful": []
```

```
}
```

- 자세한 API 내용은 명령 참조DeleteVpcEndpoints의 섹션을 참조하세요. AWS CLI

### `delete-vpc-peering-connection`

다음 코드 예시에서는 `delete-vpc-peering-connection`을 사용하는 방법을 보여 줍니다.

AWS CLI

VPC 피어링 연결을 삭제하려면

이 예제에서는 지정된 VPC 피어링 연결을 삭제합니다.

명령:

```
aws ec2 delete-vpc-peering-connection --vpc-peering-connection-id pcx-1a2b3c4d
```

출력:

```
{
    "Return": true
}
```

- 자세한 API 내용은 명령 참조DeleteVpcPeeringConnection의 섹션을 참조하세요. AWS CLI

### `delete-vpc`

다음 코드 예시에서는 `delete-vpc`을 사용하는 방법을 보여 줍니다.

AWS CLI

를 삭제하려면 VPC

이 예제에서는 지정된 를 삭제합니다VPC. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 delete-vpc --vpc-id vpc-a01106c2
```

- 자세한 API 내용은 명령 참조DeleteVpc의 섹션을 참조하세요. AWS CLI

### delete-vpn-connection-route

다음 코드 예시에서는 delete-vpn-connection-route을 사용하는 방법을 보여 줍니다.

AWS CLI

VPN 연결에서 정적 경로를 삭제하려면

이 예제에서는 지정된 VPN 연결에서 지정된 정적 경로를 삭제합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 delete-vpn-connection-route --vpn-connection-id vpn-40f41529 --destination-cidr-block 11.12.0.0/16
```

- 자세한 API 내용은 명령 참조DeleteVpnConnectionRoute의 섹션을 참조하세요. AWS CLI

### delete-vpn-connection

다음 코드 예시에서는 delete-vpn-connection을 사용하는 방법을 보여 줍니다.

AWS CLI

VPN 연결을 삭제하려면

이 예제에서는 지정된 VPN 연결을 삭제합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 delete-vpn-connection --vpn-connection-id vpn-40f41529
```

- 자세한 API 내용은 명령 참조DeleteVpnConnection의 섹션을 참조하세요. AWS CLI

### delete-vpn-gateway

다음 코드 예시에서는 delete-vpn-gateway을 사용하는 방법을 보여 줍니다.

AWS CLI

가상 프라이빗 게이트웨이를 삭제하려면

이 예제에서는 지정된 가상 프라이빗 게이트웨이를 삭제합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 delete-vpn-gateway --vpn-gateway-id vgw-9a4cacf3
```

- 자세한 API 내용은 명령 참조DeleteVpnGateway의 섹션을 참조하세요. AWS CLI

### deprovision-byoip-cidr

다음 코드 예시에서는 deprovision-byoip-cidr을 사용하는 방법을 보여 줍니다.

AWS CLI

IP 주소 범위를 사용에서 제거하려면

다음 예제에서는 에서 지정된 주소 범위를 사용하지 않도록 제거합니다 AWS.

```
aws ec2 deprovision-byoip-cidr \
    --cidr 203.0.113.25/24
```

출력:

```
{
    "ByoipCidr": {
        "Cidr": "203.0.113.25/24",
        "State": "pending-deprovision"
    }
}
```

- 자세한 API 내용은 명령 참조DeprovisionByoipCidr의 섹션을 참조하세요. AWS CLI

### deprovision-ipam-pool-cidr

다음 코드 예시에서는 deprovision-ipam-pool-cidr을 사용하는 방법을 보여 줍니다.

AWS CLI

IPAM 풀 프로비저닝을 해제하려면 CIDR

다음 `deprovision-ipam-pool-cidr` 예제에서는 IPAM 풀에 CIDR 프로비저닝된 를 프로비저닝 해제합니다.

(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
```

출력:

```
{
    "IpamPoolCidr": {
        "Cidr": "11.0.0.0/16",
        "State": "pending-deprovision"
    }
}
```

자세한 내용은 Amazon VPC IPAM 사용 설명서의 프로비저닝 해제 풀CIDRs을 참조하세요.

- 자세한 API 내용은 명령 참조DeprovisionIpamPoolCidr의 섹션을 참조하세요. AWS CLI

### `deregister-image`

다음 코드 예시에서는 `deregister-image`을 사용하는 방법을 보여 줍니다.

AWS CLI

의 등록을 취소하려면 AMI

이 예제에서는 지정된 의 등록을 취소합니다AMI. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 deregister-image --image-id ami-4fa54026
```

- 자세한 API 내용은 명령 참조DeregisterImage의 섹션을 참조하세요. AWS CLI

### deregister-instance-event-notification-attributes

다음 코드 예시에서는 deregister-instance-event-notification-attributes을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 이벤트 알림에서 모든 태그를 제거하려면

다음 deregister-instance-event-notification-attributes 예제에서는 를 로 설정하는 효과가 IncludeAllTagsOfInstance=true있는 IncludeAllTagsOfInstance를 제거합니다false.

```
aws ec2 deregister-instance-event-notification-attributes \
    --instance-tag-attribute IncludeAllTagsOfInstance=true
```

출력:

```
{
    "InstanceTagAttribute": {
        "InstanceTagKeys": [],
        "IncludeAllTagsOfInstance": true
    }
}
```

자세한 내용은 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 인스턴스에 대해 예약된 이벤트를 참조하세요.

예제 2: 이벤트 알림에서 특정 태그를 제거하려면

다음 deregister-instance-event-notification-attributes 예제에서는 이벤트 알림에 포함된 태그에서 지정된 태그를 제거합니다. 이벤트 알림에 포함된 나머지 태그를 설명하려면 를 사용합니다describe-instance-event-notification-attributes.

```
aws ec2 deregister-instance-event-notification-attributes \
    --instance-tag-attribute InstanceTagKeys="tag-key2"
```

출력:

```
{
    "InstanceTagAttribute": {
        "InstanceTagKeys": [
            "tag-key2"
        ],
        "IncludeAllTagsOfInstance": false
    }
}
```

자세한 내용은 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 인스턴스에 대해 예약된 이벤트를 참조하세요.

- 자세한 API 내용은 명령 참조DeregisterInstanceEventNotificationAttributes의 섹션을 참조하세요. AWS CLI

### deregister-transit-gateway-multicast-group-members

다음 코드 예시에서는 deregister-transit-gateway-multicast-group-members을 사용하는 방법을 보여 줍니다.

AWS CLI

멀티캐스트 그룹에서 그룹 멤버 등록을 취소하려면

이 예제에서는 전송 게이트웨이 멀티캐스트 그룹에서 지정된 네트워크 인터페이스 그룹 멤버의 등록을 취소합니다.

```
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
```

출력:

```
{
```

```
    "DeregisteredMulticastGroupMembers": {
        "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef7EXAMPLE",
        "RegisteredNetworkInterfaceIds": [
            "eni-0e246d3269EXAMPLE"
        ],
        "GroupIpAddress": "224.0.1.0"
    }
}
```

자세한 내용은 Transit Gateways 사용 설명서의 멀티캐스트 그룹에서 멤버 등록 취소를 참조하세요. AWS

- 자세한 API 내용은 명령 참조DeregisterTransitGatewayMulticastGroupMembers의 섹션을 참조하세요. AWS CLI

### deregister-transit-gateway-multicast-group-source

다음 코드 예시에서는 deregister-transit-gateway-multicast-group-source을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 멀티캐스트 그룹에서 소스 등록을 취소하려면

이 예제에서는 멀티캐스트 그룹에서 지정된 네트워크 인터페이스 그룹 소스의 등록을 취소합니다.

```
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
```

출력:

```
{
    "DeregisteredMulticastGroupSources": {
        "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef79d6e597",
        "DeregisteredNetworkInterfaceIds": [
            "eni-07f290fc3c090cbae"
        ],
        "GroupIpAddress": "224.0.1.0"
    }
```

```
}
```

자세한 내용은 Transit Gateways 사용 설명서의 멀티캐스트 그룹에서 소스 등록 취소를 참조하세요. AWS

- 자세한 API 내용은 명령 참조DeregisterTransitGatewayMulticastGroupSource의 섹션을 참조하세요. AWS CLI

### `describe-account-attributes`

다음 코드 예시에서는 `describe-account-attributes`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 계정의 모든 속성을 설명하려면

이 예제에서는 AWS 계정의 속성을 설명합니다.

명령:

```
aws ec2 describe-account-attributes
```

출력:

```
{
    "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"
                }
            ]
        }
    ]
}
```

AWS 계정의 단일 속성을 설명하려면

이 예제에서는 AWS 계정의 `supported-platforms` 속성을 설명합니다.

명령:

```
aws ec2 describe-account-attributes --attribute-names supported-platforms
```

출력:

```
{
    "AccountAttributes": [
        {
            "AttributeName": "supported-platforms",
            "AttributeValues": [
                {
                    "AttributeValue": "EC2"
                },
                {
                    "AttributeValue": "VPC"
                }
            ]
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeAccountAttributes의 섹션을 참조하세요. AWS CLI

### describe-address-transfers

다음 코드 예시에서는 describe-address-transfers을 사용하는 방법을 보여 줍니다.

AWS CLI

탄력적 IP 주소 전송을 설명하려면

다음 describe-address-transfers 예제에서는 지정된 탄력적 IP 주소에 대한 탄력적 IP 주소 전송을 설명합니다.

```
aws ec2 describe-address-transfers \
    --allocation-ids eipalloc-09ad461b0d03f6aaf
```

출력:

```
{
    "AddressTransfers": [
```

```
        {
            "PublicIp": "100.21.184.216",
            "AllocationId": "eipalloc-09ad461b0d03f6aaf",
            "TransferAccountId": "123456789012",
            "TransferOfferExpirationTimestamp": "2023-02-22T22:51:01.000Z",
            "AddressTransferStatus": "pending"
        }
    ]
}
```

자세한 내용은 Amazon VPC 사용 설명서의 탄력적 IP 주소 전송을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeAddressTransfers의 섹션을 참조하세요. AWS CLI

### describe-addresses-attribute

다음 코드 예시에서는 describe-addresses-attribute을 사용하는 방법을 보여 줍니다.

AWS CLI

탄력적 IP 주소와 연결된 도메인 이름의 속성을 보려면

다음 describe-addresses-attribute 예제에서는 탄력적 IP 주소와 연결된 도메인 이름의 속성을 반환합니다.

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
```

출력:

```
{
    "Addresses": [
```

```
        {
            "PublicIp": "192.0.2.0",
            "AllocationId": "eipalloc-abcdef01234567890",
            "PtrRecord": "example.com."
        }
    ]
}
```

탄력적 IP 주소의 속성을 보려면 먼저 도메인 이름을 탄력적 IP 주소와 연결해야 합니다. 자세한 내용은 Amazon EC2 사용 설명서 또는 AWS CLI 명령 참조modify-address-attribute의 이메일 애플리케이션에 DNS 역방향 사용을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeAddressesAttribute의 섹션을 참조하세요. AWS CLI

### `describe-addresses`

다음 코드 예시에서는 `describe-addresses`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 모든 탄력적 IP 주소에 대한 세부 정보를 검색하는 방법

다음 `describe addresses` 예제에서는 탄력적 IP 주소에 대한 세부 정보를 표시합니다.

```
aws ec2 describe-addresses
```

출력:

```
{
    "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"
        }
    ]
}
```

예제 2: EC2에 대한 탄력적 IP 주소의 세부 정보를 검색하려면VPC

다음 `describe-addresses` 예제에서는 의 인스턴스와 함께 사용할 탄력적 IP 주소에 대한 세부 정보를 보여줍니다VPC.

```
aws ec2 describe-addresses \
    --filters "Name=domain,Values=vpc"
```

출력:

```
{
    "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"
        }
    ]
}
```

예제 3: 할당 ID로 지정된 탄력적 IP 주소의 세부 정보를 검색하는 방법

다음 `describe-addresses` 예제에서는 EC2-의 인스턴스와 연결된 지정된 할당 ID가 있는 탄력적 IP 주소에 대한 세부 정보를 표시합니다VPC.

```
aws ec2 describe-addresses \
    --allocation-ids eipalloc-282d9641
```

출력:

```
{
    "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"
        }
    ]
}
```

예제 4: VPC 프라이빗 IP 주소로 지정된 탄력적 IP 주소에 대한 세부 정보를 검색하려면

다음 `describe-addresses` 예제에서는 EC2-의 특정 프라이빗 IP 주소와 연결된 탄력적 IP 주소에 대한 세부 정보를 표시합니다VPC.

```
aws ec2 describe-addresses \
    --filters "Name=private-ip-address,Values=10.251.50.12"
```

예제 5: EC2-Classic에서 탄력적 IP 주소에 대한 세부 정보를 검색하려면

TThe 다음 `describe-addresses` 예제에서는 EC2-Classic에서 사용할 탄력적 IP 주소에 대한 세부 정보를 표시합니다.

```
aws ec2 describe-addresses \
    --filters "Name=domain,Values=standard"
```

출력:

```
{
    "Addresses": [
        {
            "InstanceId": "i-1234567890abcdef0",
            "PublicIp": "203.0.110.25",
```

```
            "PublicIpv4Pool": "amazon",
            "Domain": "standard"
        }
    ]
}
```

예제 6: 퍼블릭 IP 주소로 지정된 탄력적 IP 주소의 세부 정보를 검색하는 방법

다음 `describe-addresses` 예제에서는 EC2-Classic의 인스턴스와 `203.0.110.25`연결된 값을 가진 탄력적 IP 주소에 대한 세부 정보를 표시합니다.

```
aws ec2 describe-addresses \
    --public-ips 203.0.110.25
```

출력:

```
{
    "Addresses": [
        {
            "InstanceId": "i-1234567890abcdef0",
            "PublicIp": "203.0.110.25",
            "PublicIpv4Pool": "amazon",
            "Domain": "standard"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeAddresses의 섹션을 참조하세요. AWS CLI

### `describe-aggregate-id-format`

다음 코드 예시에서는 `describe-aggregate-id-format`을 사용하는 방법을 보여 줍니다.

AWS CLI

리전의 모든 리소스 유형에 대한 더 긴 ID 형식 설정을 설명하려면

다음 `describe-aggregate-id-format` 예제에서는 현재 리전의 전체 긴 ID 형식 상태를 설명합니다. 이 `Deadline` 값은 이러한 리소스의 기한이 짧은 ID 형식에서 긴 ID 형식으로 영구적으로 전환될 때까지 만료되었음을 나타냅니다. `UseLongIdsAggregated` 값은 모든 IAM 사용자와 IAM 역할이 모든 리소스 유형에 긴 ID 형식을 사용하도록 구성되어 있음을 나타냅니다.

```
aws ec2 describe-aggregate-id-format
```

출력:

```
{
    "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
        },
        ...
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeAggregateIdFormat의 섹션을 참조하세요. AWS CLI

### describe-availability-zones

다음 코드 예시에서는 describe-availability-zones을 사용하는 방법을 보여 줍니다.

AWS CLI

가용 영역을 설명하는 방법

다음 describe-availability-zones 예제에서는 사용 가능한 가용 영역에 대한 세부 정보를 표시합니다. 응답에는 현재 리전의 가용 영역만 포함됩니다. 이 예제에서는 프로파일의 기본 us-west-2(오레곤) 리전을 사용합니다.

```
aws ec2 describe-availability-zones
```

출력:


```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeAvailabilityZones의 섹션을 참조하세요. AWS CLI

### `describe-aws-network-performance-metric-subscription`

다음 코드 예시에서는 `describe-aws-network-performance-metric-subscription`을 사용하는 방법을 보여 줍니다.

AWS CLI

지표 구독을 설명하려면

다음 `describe-aws-network-performance-metric-subscriptions` 예제에서는 지표 구독에 대해 설명합니다.

```
aws ec2 describe-aws-network-performance-metric-subscriptions
```

출력:

```
{
    "Subscriptions": [
        {
            "Source": "us-east-1",
            "Destination": "eu-west-1",
            "Metric": "aggregate-latency",
            "Statistic": "p50",
            "Period": "five-minutes"
        }
    ]
}
```

자세한 내용은 인프라 성능 사용 설명서의 구독 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeAwsNetworkPerformanceMetricSubscription의 섹션을 참조하세요. AWS CLI

### `describe-aws-network-performance-metric-subscriptions`

다음 코드 예시에서는 `describe-aws-network-performance-metric-subscriptions`을 사용하는 방법을 보여 줍니다.

AWS CLI

지표 구독을 설명하려면

다음 `describe-aws-network-performance-metric-subscriptions` 예제에서는 지표 구독에 대해 설명합니다.

```
aws ec2 describe-aws-network-performance-metric-subscriptions
```

출력:

```
{
    "Subscriptions": [
        {
            "Source": "us-east-1",
            "Destination": "eu-west-1",
            "Metric": "aggregate-latency",
            "Statistic": "p50",
            "Period": "five-minutes"
        }
    ]
}
```

자세한 내용은 인프라 성능 사용 설명서의 구독 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeAwsNetworkPerformanceMetricSubscriptions의 섹션을 참조하세요. AWS CLI

### `describe-bundle-tasks`

다음 코드 예시에서는 `describe-bundle-tasks`을 사용하는 방법을 보여 줍니다.

AWS CLI

번들 작업을 설명하려면

이 예제에서는 모든 번들 작업에 대해 설명합니다.

명령:

```
aws ec2 describe-bundle-tasks
```

출력:

```
{
  "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"
    }
  ]
}
```

- 자세한 API 내용은 명령 참조DescribeBundleTasks의 섹션을 참조하세요. AWS CLI

### `describe-byoip-cidrs`

다음 코드 예시에서는 `describe-byoip-cidrs`을 사용하는 방법을 보여 줍니다.

AWS CLI

프로비저닝된 주소 범위를 설명하려면

다음 `describe-byoip-cidrs` 예제에서는 에서 사용하도록 프로비저닝한 퍼블릭 IPv4 주소 범위에 대한 세부 정보를 보여줍니다 AWS.

```
aws ec2 describe-byoip-cidrs
```

출력:

```
{
    "ByoipCidrs": [
        {
            "Cidr": "203.0.113.25/24",
            "StatusMessage": "ipv4pool-ec2-1234567890abcdef0",
            "State": "provisioned"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeByoipCidrs의 섹션을 참조하세요. AWS CLI

### describe-capacity-reservation-fleets

다음 코드 예시에서는 describe-capacity-reservation-fleets을 사용하는 방법을 보여 줍니다.

AWS CLI

용량 예약 플릿을 보려면

다음 describe-capacity-reservation-fleets 예제에서는 지정된 용량 예약 플릿에 대한 구성 및 용량 정보를 나열합니다. 또한 플릿 내에 있는 개별 용량 예약에 대한 세부 정보도 나열됩니다.

```
aws ec2 describe-capacity-reservation-fleets \
    --capacity-reservation-fleet-ids crf-abcdef01234567890
```

출력:

```
{
    "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"
        }
    ]
}
```

용량 예약 플릿에 대한 자세한 내용은 Amazon EC2 사용 설명서의 용량 예약 플릿을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeCapacityReservationFleets의 섹션을 참조하세요. AWS CLI

### `describe-capacity-reservations`

다음 코드 예시에서는 `describe-capacity-reservations`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 용량 예약 중 하나 이상을 설명하려면

다음 `describe-capacity-reservations` 예제에서는 현재 AWS 리전의 모든 용량 예약에 대한 세부 정보를 표시합니다.

```
aws ec2 describe-capacity-reservations
```

출력:

```
{
    "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"
        }
    ]
}
```

예제 2: 용량 예약 중 하나 이상을 설명하려면

다음 `describe-capacity-reservations` 예제에서는 지정된 용량 예약에 대한 세부 정보를 표시합니다.

```
aws ec2 describe-capacity-reservations \
    --capacity-reservation-ids cr-1234abcd56EXAMPLE
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 용량 예약 보기를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeCapacityReservations의 섹션을 참조하세요. AWS CLI

### describe-carrier-gateways

다음 코드 예시에서는 describe-carrier-gateways을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 통신 사업자 게이트웨이를 설명하려면

다음 describe-carrier-gateways 예제에서는 모든 캐리어 게이트웨이를 나열합니다.

```
aws ec2 describe-carrier-gateways
```

출력:

```
{
    "CarrierGateways": [
        {
            "CarrierGatewayId": "cagw-0465cdEXAMPLE1111",
            "VpcId": "vpc-0c529aEXAMPLE",
            "State": "available",
            "OwnerId": "123456789012",
            "Tags": [
                {

                    "Key": "example",
                    "Value": "tag"
                }
            ]
        }
    ]
}
```

자세한 내용은 Amazon Virtual Private Cloud 사용 설명서의 통신 사업자 게이트웨이<https://docs.aws.amazon.com/vpc/latest/userguide/Carrier_Gateway.html>를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeCarrierGateways의 섹션을 참조하세요. AWS CLI

### `describe-classic-link-instances`

다음 코드 예시에서는 `describe-classic-link-instances`을 사용하는 방법을 보여 줍니다.

AWS CLI

연결된 EC2-Classic 인스턴스를 설명하려면

이 예제에서는 연결된 모든 EC2-Classic 인스턴스를 나열합니다.

명령:

```
aws ec2 describe-classic-link-instances
```

출력:

```
{
        "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"
                                }
                        ]
                }
        ]
}
```

이 예제에서는 연결된 모든 EC2-Classic 인스턴스를 나열하고 VPC vpc-88888888에 연결된 인스턴스만 포함하도록 응답을 필터링합니다.

명령:

```
aws ec2 describe-classic-link-instances --filter "Name=vpc-id,Values=vpc-88888888"
```

출력:

```
{
```

```
        "Instances": [
                {
                        "InstanceId": "i-1234567890abcdef0",
                        "VpcId": "vpc-88888888",
                        "Groups": [
                                {
                                        "GroupId": "sg-11122233"
                                }
                        ],
                        "Tags": [
                                {
                                        "Value": "ClassicInstance",
                                        "Key": "Name"
                                }
                        ]
                }
        ]
}
```

- 자세한 API 내용은 명령 참조DescribeClassicLinkInstances의 섹션을 참조하세요. AWS CLI

### `describe-client-vpn-authorization-rules`

다음 코드 예시에서는 `describe-client-vpn-authorization-rules`을 사용하는 방법을 보여 줍니다.

AWS CLI

클라이언트 VPN 엔드포인트에 대한 권한 부여 규칙을 설명하려면

다음 `describe-client-vpn-authorization-rules` 예제에서는 지정된 클라이언트 VPN 엔드포인트의 권한 부여 규칙에 대한 세부 정보를 표시합니다.

```
aws ec2 describe-client-vpn-authorization-rules \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

출력:

```
{
    "AuthorizationRules": [
        {
            "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",
```

```
            "GroupId": "",
            "AccessAll": true,
            "DestinationCidr": "0.0.0.0/0",
            "Status": {
                "Code": "active"
            }
        }
    ]
}
```

자세한 내용은 AWS 클라이언트 VPN 관리자 안내서의 권한 부여 규칙을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeClientVpnAuthorizationRules의 섹션을 참조하세요. AWS CLI

### describe-client-vpn-connections

다음 코드 예시에서는 describe-client-vpn-connections을 사용하는 방법을 보여 줍니다.

AWS CLI

클라이언트 VPN 엔드포인트에 대한 연결을 설명하려면

다음 describe-client-vpn-connections 예제에서는 지정된 클라이언트 VPN 엔드포인트에 대한 클라이언트 연결에 대한 세부 정보를 표시합니다.

```
aws ec2 describe-client-vpn-connections \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

출력:

```
{
    "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": "-"
        }
    ]
}
```

자세한 내용은 클라이언트 관리자 안내서의 클라이언트 연결을 참조하세요. AWS VPN

- 자세한 API 내용은 명령 참조DescribeClientVpnConnections의 섹션을 참조하세요. AWS CLI

### `describe-client-vpn-endpoints`

다음 코드 예시에서는 `describe-client-vpn-endpoints`을 사용하는 방법을 보여 줍니다.

AWS CLI

클라이언트 VPN 엔드포인트를 설명하려면

다음 `describe-client-vpn-endpoints` 예제에서는 모든 Client VPN 엔드포인트에 대한 세부 정보를 표시합니다.

```
aws ec2 describe-client-vpn-endpoints
```

출력:

```
{
    "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
            }
        }
    ]
}
```

자세한 내용은 클라이언트 관리자 안내서의 클라이언트 VPN 엔드포인트를 참조하세요. AWS VPN

- 자세한 API 내용은 명령 참조DescribeClientVpnEndpoints의 섹션을 참조하세요. AWS CLI

### `describe-client-vpn-routes`

다음 코드 예시에서는 `describe-client-vpn-routes`을 사용하는 방법을 보여 줍니다.

AWS CLI

클라이언트 VPN 엔드포인트의 경로를 설명하려면

다음 `describe-client-vpn-routes` 예제에서는 지정된 클라이언트 VPN 엔드포인트의 경로에 대한 세부 정보를 표시합니다.

```
aws ec2 describe-client-vpn-routes \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

출력:

```
{
    "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"
            }
        }
    ]
}
```

자세한 내용은 AWS 클라이언트 VPN 관리자 안내서의 라우팅을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeClientVpnRoutes의 섹션을 참조하세요. AWS CLI

### describe-client-vpn-target-networks

다음 코드 예시에서는 describe-client-vpn-target-networks을 사용하는 방법을 보여 줍니다.

AWS CLI

클라이언트 VPN 엔드포인트의 대상 네트워크를 설명하려면

다음 describe-client-vpn-target-networks 예제에서는 지정된 클라이언트 VPN 엔드포인트의 대상 네트워크에 대한 세부 정보를 표시합니다.

```
aws ec2 describe-client-vpn-target-networks \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

출력:

```
{
    "ClientVpnTargetNetworks": [
        {
            "AssociationId": "cvpn-assoc-012e837060753dc3d",
```

```
            "VpcId": "vpc-11111222222333333",
            "TargetNetworkId": "subnet-0123456789abcabca",
            "ClientVpnEndpointId": "cvpn-endpoint-123456789123abcde",
            "Status": {
                "Code": "associating"
            },
            "SecurityGroups": [
                "sg-012345678910abcab"
            ]
        }
    ]
}
```

자세한 내용은 AWS 클라이언트 VPN 관리자 안내서의 대상 네트워크를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeClientVpnTargetNetworks의 섹션을 참조하세요. AWS CLI

### describe-coip-pools

다음 코드 예시에서는 describe-coip-pools을 사용하는 방법을 보여 줍니다.

AWS CLI

고객 소유 IP 주소 풀을 설명하려면

다음 describe-coip-pools 예제에서는 AWS 계정의 고객 소유 IP 주소 풀을 설명합니다.

```
aws ec2 describe-coip-pools
```

출력:

```
{
    "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"
        }
```

```
    ]
}
```

자세한 내용은 AWS Outposts 사용 설명서의 고객 소유 IP 주소를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeCoipPools의 섹션을 참조하세요. AWS CLI

### describe-conversion-tasks

다음 코드 예시에서는 describe-conversion-tasks을 사용하는 방법을 보여 줍니다.

AWS CLI

변환 작업의 상태를 보려면

이 예제에서는 ID가 import-i-ffvko9js인 변환 작업의 상태를 반환합니다.

명령:

```
aws ec2 describe-conversion-tasks --conversion-task-ids import-i-ffvko9js
```

출력:

```
{
    "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"
    }
]
}
```

- 자세한 API 내용은 명령 참조DescribeConversionTasks의 섹션을 참조하세요. AWS CLI

### `describe-customer-gateways`

다음 코드 예시에서는 `describe-customer-gateways`을 사용하는 방법을 보여 줍니다.

AWS CLI

고객 게이트웨이를 설명하려면

이 예제에서는 고객 게이트웨이에 대해 설명합니다.

명령:

```
aws ec2 describe-customer-gateways
```

출력:

```
{
    "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"
        }
    ]
}
```

특정 고객 게이트웨이를 설명하려면

이 예제에서는 지정된 고객 게이트웨이를 설명합니다.

명령:

```
aws ec2 describe-customer-gateways --customer-gateway-ids cgw-0e11f167
```

출력:

```
{
    "CustomerGateways": [
        {
            "CustomerGatewayId": "cgw-0e11f167",
            "IpAddress": "12.1.2.3",
            "State": "available",
            "Type": "ipsec.1",
            "BgpAsn": "65534"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeCustomerGateways의 섹션을 참조하세요. AWS CLI

### describe-dhcp-options

다음 코드 예시에서는 describe-dhcp-options을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: DHCP 옵션 설명

다음 describe-dhcp-options 예제에서는 DHCP 옵션에 대한 세부 정보를 검색합니다.

```
aws ec2 describe-dhcp-options
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS VPC 사용 설명서의 DHCP 옵션 세트 작업을 참조하세요.

예제 2: DHCP 옵션을 설명하고 출력을 필터링하려면

다음 `describe-dhcp-options` 예제에서는 DHCP 옵션을 설명하고 필터를 사용하여 도메인 이름 서버에 `example.com` 대해 가 있는 DHCP 옵션만 반환합니다. 이 예제에서는 `--query` 파라미터를 사용하여 출력에 구성 정보와 ID만 표시합니다.

```
aws ec2 describe-dhcp-options \
    --filters Name=key,Values=domain-name-servers Name=value,Values=example.com \
    --query "DhcpOptions[*].[DhcpConfigurations,DhcpOptionsId]"
```

출력:

```
[
    [
        [
            {
                "Key": "domain-name",
                "Values": [
                    {
                        "Value": "example.com"
                    }
                ]
            },
            {
                "Key": "domain-name-servers",
                "Values": [
                    {
                        "Value": "172.16.16.16"
                    }
                ]
            }
```

```
        ],
        "dopt-001122334455667ab"
    ]
]
```

자세한 내용은 AWS VPC 사용 설명서의 DHCP 옵션 세트 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeDhcpOptions의 섹션을 참조하세요. AWS CLI

### `describe-egress-only-internet-gateways`

다음 코드 예시에서는 `describe-egress-only-internet-gateways`을 사용하는 방법을 보여 줍니다.

AWS CLI

송신 전용 인터넷 게이트웨이를 설명하려면

이 예제에서는 송신 전용 인터넷 게이트웨이에 대해 설명합니다.

명령:

```
aws ec2 describe-egress-only-internet-gateways
```

출력:

```
{
  "EgressOnlyInternetGateways": [
      {
          "EgressOnlyInternetGatewayId": "eigw-015e0e244e24dfe8a",
          "Attachments": [
              {
                  "State": "attached",
                  "VpcId": "vpc-0c62a468"
              }
          ]
      }
  ]
}
```

- 자세한 API 내용은 명령 참조DescribeEgressOnlyInternetGateways의 섹션을 참조하세요. AWS CLI

### describe-elastic-gpus

다음 코드 예시에서는 describe-elastic-gpus을 사용하는 방법을 보여 줍니다.

AWS CLI

탄력적 GPU

명령:

```
aws ec2 describe-elastic-gpus --elastic-gpu-
ids egpu-12345678901234567890abcdefghijkl
```

- 자세한 API 내용은 명령 참조DescribeElasticGpus의 섹션을 참조하세요. AWS CLI

### describe-export-image-tasks

다음 코드 예시에서는 describe-export-image-tasks을 사용하는 방법을 보여 줍니다.

AWS CLI

이미지 내보내기 작업을 모니터링하려면

다음 describe-export-image-tasks 예제에서는 지정된 내보내기 이미지 작업의 상태를 확인합니다. Amazon S3의 결과 이미지 파일은 입니다my-export-bucket/exports/export-ami-1234567890abcdef0.vmdk.

```
aws ec2 describe-export-image-tasks \
    --export-image-task-ids export-ami-1234567890abcdef0
```

진행 중인 이미지 내보내기 작업의 출력입니다.

```
{
    "ExportImageTasks": [
        {
            "ExportImageTaskId": "export-ami-1234567890abcdef0"
            "Progress": "21",
            "S3ExportLocation": {
                "S3Bucket": "my-export-bucket",
                "S3Prefix": "exports/"
            },
            "Status": "active",
```

```
            "StatusMessage": "updating"
        }
    ]
}
```

완료된 이미지 내보내기 작업의 출력입니다.

```
{
    "ExportImageTasks": [
        {
            "ExportImageTaskId": "export-ami-1234567890abcdef0"
            "S3ExportLocation": {
                "S3Bucket": "my-export-bucket",
                "S3Prefix": "exports/"
            },
            "Status": "completed"
        }
    ]
}
```

자세한 내용은 VM 가져오기/내보내기 사용 설명서AMI의 에서 VM 내보내기를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeExportImageTasks의 섹션을 참조하세요. AWS CLI

### describe-export-tasks

다음 코드 예시에서는 describe-export-tasks을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스 내보내기 작업에 대한 세부 정보를 나열하려면

이 예제에서는 ID가 export-i-fh8sjjsq인 내보내기 작업에 대해 설명합니다.

명령:

```
aws ec2 describe-export-tasks --export-task-ids export-i-fh8sjjsq
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeExportTasks의 섹션을 참조하세요. AWS CLI

### `describe-fast-launch-images`

다음 코드 예시에서는 `describe-fast-launch-images`을 사용하는 방법을 보여 줍니다.

AWS CLI

더 빠른 시작을 위해 AMIs 구성된 Windows에 대한 세부 정보를 설명하려면

다음 `describe-fast-launch-images` 예제에서는 리소스 유형, 스냅샷 구성, 시작 템플릿 세부 정보, 최대 병렬 시작 수, AMI 소유자 ID, 빠른 시작 구성 상태, 상태가 변경된 이유, 상태 변경이 발생한 시간을 포함하여 더 빠른 시작을 위해 구성된 AMIs 계정의 각 에 대한 세부 정보를 설명합니다.

```
aws ec2 describe-fast-launch-images
```

출력:

```
{
    "FastLaunchImages": [
        {
            "ImageId": "ami-01234567890abcedf",
            "ResourceType": "snapshot",
            "SnapshotConfiguration": {},
```

```
            "LaunchTemplate": {
                "LaunchTemplateId": "lt-01234567890abcedf",
                "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"
        }
    ]
}
```

더 빠른 시작을 AMI 위해 Windows를 구성하는 방법에 대한 자세한 내용은 Amazon EC2 사용 설명서 의 더 빠른 시작을 AMI 위해 를 구성하는 단원을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeFastLaunchImages의 섹션을 참조하세요. AWS CLI

### describe-fast-snapshot-restores

다음 코드 예시에서는 describe-fast-snapshot-restores을 사용하는 방법을 보여 줍니다.

AWS CLI

빠른 스냅샷 복원을 설명하려면

다음 describe-fast-snapshot-restores 예제에서는 상태가 인 모든 빠른 스냅샷 복원에 대한 세부 정보를 표시합니다disabled.

```
aws ec2 describe-fast-snapshot-restores \
    --filters Name=state,Values=disabled
```

출력:

```
{
    "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"
        }
    ]
}
```

다음 describe-fast-snapshot-restores 예제에서는 모든 빠른 스냅샷 복원을 설명합니다.

```
aws ec2 describe-fast-snapshot-restores
```

- 자세한 API 내용은 명령 참조DescribeFastSnapshotRestores의 섹션을 참조하세요. AWS CLI

### describe-fleet-history

다음 코드 예시에서는 describe-fleet-history을 사용하는 방법을 보여 줍니다.

AWS CLI

EC2플릿 기록을 설명하려면

다음 describe-fleet-history 예제에서는 지정된 시간에 시작하는 지정된 EC2 플릿에 대한 기록을 반환합니다. 출력은 실행 중인 인스턴스가 두 개 있는 EC2 플릿에 대한 것입니다.

```
aws ec2 describe-fleet-history \
    --fleet-id fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE \
    --start-time 2020-09-01T00:00:00Z
```

출력:

```
{
    "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"
}
```

자세한 내용은 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 EC2 플릿 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeFleetHistory의 섹션을 참조하세요. AWS CLI

### `describe-fleet-instances`

다음 코드 예시에서는 `describe-fleet-instances`을 사용하는 방법을 보여 줍니다.

AWS CLI

EC2플릿에 대해 실행 중인 인스턴스를 설명하려면

다음 `describe-fleet-instances` 예제에서는 지정된 EC2 플릿에 대해 실행 중인 인스턴스를 설명합니다.

```
aws ec2 describe-fleet-instances \
    --fleet-id 12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE
```

출력:

```
{
    "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"
}
```

자세한 내용은 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 EC2 플릿 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeFleetInstances의 섹션을 참조하세요. AWS CLI

### describe-fleets

다음 코드 예시에서는 describe-fleets을 사용하는 방법을 보여 줍니다.

AWS CLI

EC2플릿을 설명하려면

다음 describe-fleets 예제에서는 지정된 EC2 플릿을 설명합니다.

```
aws ec2 describe-fleets \
    --fleet-ids fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE
```

출력:

```
{
    "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"
            }
        }
    ]
}
```

자세한 내용은 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 EC2 플릿 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeFleets의 섹션을 참조하세요. AWS CLI

### describe-flow-logs

다음 코드 예시에서는 describe-flow-logs을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 모든 흐름 로그 설명

다음 describe-flow-logs 예제에서는 모든 흐름 로그에 대한 세부 정보를 표시합니다.

```
aws ec2 describe-flow-logs
```

출력:

```
{
    "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}"
        }
    ]
}
```

예제 2: 흐름 로그의 하위 집합 설명

다음 `describe-flow-logs` 예제에서는 필터를 사용하여 Amazon CloudWatch Logs에서 지정된 로그 그룹에 있는 흐름 로그에 대한 세부 정보만 표시합니다.

```
aws ec2 describe-flow-logs \
    --filter "Name=log-group-name,Values=MyFlowLogs"
```

- 자세한 API 내용은 명령 참조DescribeFlowLogs의 섹션을 참조하세요. AWS CLI

### `describe-fpga-image-attribute`

다음 코드 예시에서는 `describe-fpga-image-attribute`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon FPGA 이미지의 속성을 설명하려면

이 예제에서는 지정된 에 대한 로드 권한을 설명합니다AFI.

명령:

```
aws ec2 describe-fpga-image-attribute --fpga-image-id afi-0d123e123bfc85abc --
attribute loadPermission
```

출력:

```
{
  "FpgaImageAttribute": {
      "FpgaImageId": "afi-0d123e123bfc85abc",
      "LoadPermissions": [
          {
              "UserId": "123456789012"
          }
      ]
  }
}
```

- 자세한 API 내용은 명령 참조DescribeFpgaImageAttribute의 섹션을 참조하세요. AWS CLI

### describe-fpga-images

다음 코드 예시에서는 describe-fpga-images을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon FPGA 이미지를 설명하려면

이 예제에서는 계정 에서 소유AFIs한 를 설명합니다123456789012.

명령:

```
aws ec2 describe-fpga-images --filters Name=owner-id,Values=123456789012
```

출력:

```
{
  "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeFpgaImages의 섹션을 참조하세요. AWS CLI

### describe-host-reservation-offerings

다음 코드 예시에서는 describe-host-reservation-offerings을 사용하는 방법을 보여 줍니다.

AWS CLI

전용 호스트 예약 서비스를 설명하려면

이 예제에서는 구매할 수 있는 M4 인스턴스 패밀리의 전용 호스트 예약에 대해 설명합니다.

명령:

```
aws ec2 describe-host-reservation-offerings --filter Name=instance-family,Values=m4
```

출력:

```
{
    "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
        }
    ]
```

```
}
```

- 자세한 API 내용은 명령 참조DescribeHostReservationOfferings의 섹션을 참조하세요. AWS CLI

### `describe-host-reservations`

다음 코드 예시에서는 `describe-host-reservations`을 사용하는 방법을 보여 줍니다.

AWS CLI

계정의 전용 호스트 예약을 설명하려면

이 예제에서는 계정의 전용 호스트 예약에 대해 설명합니다.

명령:

```
aws ec2 describe-host-reservations
```

출력:

```
{
  "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
      }
  ]
}
```

- 자세한 API 내용은 명령 참조DescribeHostReservations의 섹션을 참조하세요. AWS CLI

### describe-hosts

다음 코드 예시에서는 describe-hosts을 사용하는 방법을 보여 줍니다.

AWS CLI

전용 호스트에 대한 세부 정보를 보려면

다음 describe-hosts 예제에서는 AWS 계정의 available 전용 호스트에 대한 세부 정보를 표시합니다.

```
aws ec2 describe-hosts --filter "Name=state,Values=available"
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 전용 호스트 보기를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeHosts의 섹션을 참조하세요. AWS CLI

### describe-iam-instance-profile-associations

다음 코드 예시에서는 describe-iam-instance-profile-associations을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 인스턴스 프로파일 연결을 설명하려면

이 예제에서는 모든 IAM 인스턴스 프로파일 연결을 설명합니다.

명령:

```
aws ec2 describe-iam-instance-profile-associations
```

출력:

```
{
  "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"
            }
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeIamInstanceProfileAssociations의 섹션을 참조하세요. AWS CLI

### `describe-id-format`

다음 코드 예시에서는 `describe-id-format`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 리소스의 ID 형식을 설명하려면

다음 `describe-id-format` 예제에서는 보안 그룹의 ID 형식을 설명합니다.

```
aws ec2 describe-id-format \
    --resource security-group
```

다음 예제 출력에서 `Deadline` 값은 이 리소스 유형의 기한이 짧은 ID 형식에서 긴 ID 형식으로 영구적으로 전환되도록 2018년 8월 15일 00:00UTC에 만료되었음을 나타냅니다.

```
{
    "Statuses": [
        {
            "Deadline": "2018-08-15T00:00:00.000Z",
            "Resource": "security-group",
            "UseLongIds": true
        }
    ]
}
```

예제 2: 모든 리소스의 ID 형식을 설명하려면

다음 describe-id-format 예제에서는 모든 리소스 유형의 ID 형식을 설명합니다. 짧은 ID 형식을 지원하는 모든 리소스 유형은 긴 ID 형식을 사용하도록 전환되었습니다.

```
aws ec2 describe-id-format
```

- 자세한 API 내용은 명령 참조DescribeIdFormat의 섹션을 참조하세요. AWS CLI

### describe-identity-id-format

다음 코드 예시에서는 describe-identity-id-format을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 역할의 ID 형식을 설명하려면

다음 describe-identity-id-format 예제에서는 EC2Role AWS 계정의 IAM 역할에서 생성한 인스턴스에서 수신한 ID 형식을 설명합니다.

```
aws ec2 describe-identity-id-format \
    --principal-arn arn:aws:iam::123456789012:role/my-iam-role \
    --resource instance
```

다음 출력은 이 역할에서 생성된 인스턴스가 IDs 긴 ID 형식으로 수신됨을 나타냅니다.

```
{
    "Statuses": [
        {
            "Deadline": "2016-12-15T00:00:00Z",
            "Resource": "instance",
            "UseLongIds": true
        }
    ]
}
```

IAM 사용자의 ID 형식을 설명하려면

다음 describe-identity-id-format 예제에서는 AdminUser AWS 계정의 IAM 사용자가 생성한 스냅샷에서 수신한 ID 형식을 설명합니다.

```
aws ec2 describe-identity-id-format \
    --principal-arn arn:aws:iam::123456789012:user/AdminUser \
```

```
    --resource snapshot
```

출력은 이 사용자가 생성한 스냅샷이 IDs 긴 ID 형식으로 수신됨을 나타냅니다.

```
{
    "Statuses": [
        {
            "Deadline": "2016-12-15T00:00:00Z",
            "Resource": "snapshot",
            "UseLongIds": true
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeIdentityIdFormat의 섹션을 참조하세요. AWS CLI

### `describe-image-attribute`

다음 코드 예시에서는 `describe-image-attribute`을 사용하는 방법을 보여 줍니다.

AWS CLI

에 대한 시작 권한을 설명하려면 AMI

이 예제에서는 지정된 에 대한 시작 권한을 설명합니다AMI.

명령:

```
aws ec2 describe-image-attribute --image-id ami-5731123e --
attribute launchPermission
```

출력:

```
{
    "LaunchPermissions": [
        {
            "UserId": "123456789012"
        }
    ],
    "ImageId": "ami-5731123e",
}
```

의 제품 코드를 설명하려면 AMI

이 예제에서는 지정된 의 제품 코드를 설명합니다AMI. AMI 여기에는 제품 코드가 없습니다.

명령:

```
aws ec2 describe-image-attribute --image-id ami-5731123e --attribute productCodes
```

출력:

```
{
    "ProductCodes": [],
    "ImageId": "ami-5731123e",
}
```

- 자세한 API 내용은 명령 참조DescribeImageAttribute의 섹션을 참조하세요. AWS CLI

### describe-images

다음 코드 예시에서는 describe-images을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 를 설명하려면 AMI

다음 describe-images 예제에서는 지정된 리전AMI에 지정된 를 설명합니다.

```
aws ec2 describe-images \
    --region us-east-1 \
    --image-ids ami-1234567890EXAMPLE
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon Machine Images(AMI)를 참조하세요. EC2

예제 2: 필터를 AMIs 기반으로 설명

다음 `describe-images` 예제에서는 Amazon 에서 지원하는 AmazonAMIs에서 제공하는 Windows에 대해 설명합니다EBS.

```
aws ec2 describe-images \
    --owners amazon \
    --filters "Name=platform,Values=windows" "Name=root-device-type,Values=ebs"
```

`describe-images` 출력 예제는 예제 1을 참조하세요.

필터를 사용하는 추가 예제는 Amazon EC2 사용 설명서의 리소스 나열 및 필터링을 참조하세요.

예제 3: 태그 AMIs 기반 설명

다음 `describe-images` 예제에서는 태그AMIs가 있는 모든 를 설명합니다`Type=Custom`. 이 예제에서는 `--query` 파라미터를 사용하여 AMI 만 표시합니다IDs.

```
aws ec2 describe-images \
    --filters "Name=tag:Type,Values=Custom" \
    --query 'Images[*].[ImageId]' \
    --output text
```

출력:

```
ami-1234567890EXAMPLE
ami-0abcdef1234567890
```

태그 필터를 사용하는 추가 예제는 Amazon EC2 사용 설명서의 태그 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeImages의 섹션을 참조하세요. AWS CLI

### describe-import-image-tasks

다음 코드 예시에서는 `describe-import-image-tasks`을 사용하는 방법을 보여 줍니다.

AWS CLI

이미지 가져오기 작업을 모니터링하려면

다음 `describe-import-image-tasks` 예제에서는 지정된 이미지 가져오기 작업의 상태를 확인합니다.

```
aws ec2 describe-import-image-tasks \
    --import-task-ids import-ami-1234567890abcdef0
```

진행 중인 이미지 가져오기 작업의 출력입니다.

```
{
    "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"
        }
    ]
}
```

완료된 이미지 가져오기 작업의 출력입니다. 결과 ID는 에서 AMI 제공합니다`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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeImportImageTasks의 섹션을 참조하세요. AWS CLI

### describe-import-snapshot-tasks

다음 코드 예시에서는 describe-import-snapshot-tasks을 사용하는 방법을 보여 줍니다.

AWS CLI

스냅샷 가져오기 작업을 모니터링하려면

다음 describe-import-snapshot-tasks 예제에서는 지정된 가져오기 스냅샷 작업의 상태를 확인합니다.

```
aws ec2 describe-import-snapshot-tasks \
    --import-task-ids import-snap-1234567890abcdef0
```

진행 중인 스냅샷 가져오기 작업의 출력:

```
{
    "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"
                }
            }
        }
    ]
}
```

완료된 스냅샷 가져오기 작업의 출력입니다. 결과 스냅샷의 ID는 에서 제공합니다SnapshotId.

```
{
    "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"
                }
            }
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeImportSnapshotTasks의 섹션을 참조하세요. AWS CLI

### `describe-instance-attribute`

다음 코드 예시에서는 `describe-instance-attribute`을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스 유형을 설명하려면

이 예제에서는 지정된 인스턴스의 인스턴스 유형을 설명합니다.

명령:

```
aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --
attribute instanceType
```

출력:

```
{
    "InstanceId": "i-1234567890abcdef0"
    "InstanceType": {
        "Value": "t1.micro"
    }
}
```

disableApiTermination 속성을 설명하려면

이 예제에서는 지정된 인스턴스의 `disableApiTermination` 속성을 설명합니다.

명령:

```
aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --
attribute disableApiTermination
```

출력:

```
{
"InstanceId": "i-1234567890abcdef0"
    "DisableApiTermination": {
        "Value": "false"
    }
}
```

인스턴스의 블록 디바이스 매핑을 설명하려면

이 예제에서는 지정된 인스턴스의 `blockDeviceMapping` 속성을 설명합니다.

명령:

```
aws ec2 describe-instance-attribute --instance-id i-1234567890abcdef0 --
attribute blockDeviceMapping
```

출력:

```
{
    "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"
            }
        }
    ],
}
```

- 자세한 API 내용은 명령 참조DescribeInstanceAttribute의 섹션을 참조하세요. AWS CLI

### `describe-instance-connect-endpoints`

다음 코드 예시에서는 `describe-instance-connect-endpoints`을 사용하는 방법을 보여 줍니다.

AWS CLI

EC2 인스턴스 연결 엔드포인트를 설명하려면

다음 `describe-instance-connect-endpoints` 예제에서는 지정된 EC2 Instance Connect 엔드포인트를 설명합니다.

```
aws ec2 describe-instance-connect-endpoints \
    --region us-east-1 \
    --instance-connect-endpoint-ids eice-0123456789example
```

출력:

```
{
    "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": []
        }
    ]
}
```

자세한 내용은 Amazon EC2 사용 설명서의 EC2 인스턴스 연결 엔드포인트 생성을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeInstanceConnectEndpoints의 섹션을 참조하세요. AWS CLI

### describe-instance-credit-specifications

다음 코드 예시에서는 describe-instance-credit-specifications을 사용하는 방법을 보여 줍니다.

AWS CLI

CPU 하나 이상의 인스턴스 사용에 대한 크레딧 옵션을 설명하려면

다음 describe-instance-credit-specifications 예제에서는 지정된 인스턴스의 CPU 크레딧 옵션을 설명합니다.

```
aws ec2 describe-instance-credit-specifications \
    --instance-ids i-1234567890abcdef0
```

출력:

```
{
    "InstanceCreditSpecifications": [
        {
            "InstanceId": "i-1234567890abcdef0",
            "CpuCredits": "unlimited"
        }
    ]
}
```

자세한 내용은 Amazon EC2 사용 설명서의 버스트 가능한 성능 인스턴스 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeInstanceCreditSpecifications의 섹션을 참조하세요. AWS CLI

### describe-instance-event-notification-attributes

다음 코드 예시에서는 describe-instance-event-notification-attributes을 사용하는 방법을 보여 줍니다.

AWS CLI

예약된 이벤트 알림의 태그를 설명하려면

다음 describe-instance-event-notification-attributes 예제에서는 예약된 이벤트 알림에 표시할 태그를 설명합니다.

```
aws ec2 describe-instance-event-notification-attributes
```

출력:

```
{
    "InstanceTagAttribute": {
        "InstanceTagKeys": [],
        "IncludeAllTagsOfInstance": true
    }
}
```

자세한 내용은 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 인스턴스에 대해 예약된 이벤트를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeInstanceEventNotificationAttributes의 섹션을 참조하세요. AWS CLI

### describe-instance-event-windows

다음 코드 예시에서는 describe-instance-event-windows을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 모든 이벤트 기간을 설명하려면

다음 describe-instance-event-windows 예제에서는 지정된 리전의 모든 이벤트 기간을 설명합니다.

```
aws ec2 describe-instance-event-windows \
    --region us-east-1
```

출력:

```
{
    "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"
}
```

예제 2: 특정 이벤트 기간을 설명하려면

다음 describe-instance-event-windows 예제에서는 instance-event-window 파라미터를 사용하여 특정 이벤트 기간을 설명합니다.

```
aws ec2 describe-instance-event-windows \
    --region us-east-1 \
    --instance-event-window-ids iew-0abcdef1234567890
```

출력:

```
{
    "InstanceEventWindows": [
        {
            "InstanceEventWindowId": "iew-0abcdef1234567890",
            "Name": "myEventWindowName",
            "CronExpression": "* 21-23 * * 2,3",
            "AssociationTarget": {
                "InstanceIds": [
                    "i-1234567890abcdef0",
                    "i-0598c7d356eba48d7"
                ],
                "Tags": [],
                "DedicatedHostIds": []
            },
            "State": "active",
            "Tags": []
        }
}
```

예제 3: 하나 이상의 필터와 일치하는 이벤트 기간을 설명하려면

다음 `describe-instance-event-windows` 예제에서는 `filter` 파라미터를 사용하여 하나 이상의 필터와 일치하는 이벤트 기간에 대해 설명합니다. `instance-id` 필터는 지정된 인스턴스와 연결된 모든 이벤트 기간을 설명하는 데 사용됩니다. 필터를 사용하면 직접 일치를 수행합니다. 그러나 `instance-id` 필터는 다릅니다. 인스턴스 ID와 직접 일치하지 않는 경우 인스턴스의 태그 또는 전용 호스트 ID(인스턴스가 전용 호스트인 경우)와 같은 이벤트 창과의 간접 연결로 돌아갑니다.

```
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>
```

출력:

```
{
    "InstanceEventWindows": [
        {
            "InstanceEventWindowId": "iew-0dbc0adb66f235982",
            "TimeRanges": [
                {
```

```
                    "StartWeekDay": "sunday",
                    "StartHour": 2,
                    "EndWeekDay": "sunday",
                    "EndHour": 8
                }
            ],
            "Name": "myEventWindowName",
            "AssociationTarget": {
                "InstanceIds": [],
                "Tags": [],
                "DedicatedHostIds": [
                    "h-0140d9a7ecbd102dd"
                ]
            },
            "State": "active",
            "Tags": []
        }
    ]
}
```

예제 출력에서 인스턴스는 이벤트 창과 연결된 전용 호스트에 있습니다.

이벤트 기간 제약 조건은 Amazon EC2 사용 설명서의 고려 사항을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeInstanceEventWindows의 섹션을 참조하세요. AWS CLI

### `describe-instance-status`

다음 코드 예시에서는 `describe-instance-status`을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스 상태를 설명하는 방법

다음 `describe-instance-status` 예제에서는 지정된 인스턴스의 현재 상태를 설명합니다.

```
aws ec2 describe-instance-status \
    --instance-ids i-1234567890abcdef0
```

출력:

```
{
    "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"
                    }
                ]
            }
        }
    ]
}
```

자세한 내용은 Amazon EC2 사용 설명서의 인스턴스 상태 모니터링을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeInstanceStatus의 섹션을 참조하세요. AWS CLI

### `describe-instance-topology`

다음 코드 예시에서는 `describe-instance-topology`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 인스턴스의 인스턴스 토폴로지를 설명하려면

다음 `describe-instance-topology` 예제에서는 이 명령에 지원되는 인스턴스 유형과 일치하는 모든 인스턴스의 토폴로지를 설명합니다.

```
aws ec2 describe-instance-topology \
    --region us-west-2
```

출력:

```
{
    "Instances": [
        {
            "InstanceId": "i-1111111111example",
            "InstanceType": "p4d.24xlarge",
            "GroupName": "my-ml-cpg",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3333333333example"
            ],
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        },
        {
            "InstanceId": "i-2222222222example",
            "InstanceType": "p4d.24xlarge",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-2222222222example",
                "nn-3333333333example"
            ],
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        },
        {
            "InstanceId": "i-3333333333example",
            "InstanceType": "trn1.32xlarge",
            "NetworkNodes": [
                "nn-1212121212example",
                "nn-1211122211example",
                "nn-1311133311example"
            ],
            "ZoneId": "usw2-az4",
            "AvailabilityZone": "us-west-2d"
        },
        {
            "InstanceId": "i-444444444example",
```

```
            "InstanceType": "trn1.2xlarge",
            "NetworkNodes": [
                "nn-1111111111example",
                "nn-5434334334example",
                "nn-1235301234example"
            ],
            "ZoneId": "usw2-az2",
            "AvailabilityZone": "us-west-2a"
        }
    ],
    "NextToken": "SomeEncryptedToken"
}
```

추가 예제를 포함한 자세한 내용은 Amazon 사용 설명서의 Amazon EC2 인스턴스 토폴로지를 참조하세요.  EC2

- 자세한 API 내용은 명령 참조DescribeInstanceTopology의 섹션을 참조하세요. AWS CLI

### `describe-instance-type-offerings`

다음 코드 예시에서는 `describe-instance-type-offerings`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 리전에서 제공되는 인스턴스 유형을 나열하려면

다음 `describe-instance-type-offerings` 예제에서는 의 기본 리전으로 구성된 리전에서 제공되는 인스턴스 유형을 나열합니다 AWS CLI.

```
aws ec2 describe-instance-type-offerings
```

다른 리전에서 제공되는 인스턴스 유형을 나열하려면 `--region` 파라미터를 사용하여 리전을 지정합니다.

```
aws ec2 describe-instance-type-offerings \
    --region us-east-2
```

출력:

```
{
  "InstanceTypeOfferings": [
      {
```

```
            "InstanceType": "m5.2xlarge",
            "LocationType": "region",
            "Location": "us-east-2"
        },
        {
            "InstanceType": "t3.micro",
            "LocationType": "region",
            "Location": "us-east-2"
        },
        ...
    ]
}
```

예제 2: 가용 영역에서 제공되는 인스턴스 유형을 나열하려면

다음 `describe-instance-type-offerings` 예제에서는 지정된 가용 영역에서 제공되는 인스턴스 유형을 나열합니다. 가용 영역은 지정된 리전에 있어야 합니다.

```
aws ec2 describe-instance-type-offerings \
    --location-type availability-zone \
    --filters Name=location,Values=us-east-2a \
    --region us-east-2
```

예제 3: 인스턴스 유형이 지원되는지 확인하려면

다음 `describe-instance-type-offerings` 명령은 지정된 리전에서 `c5.xlarge` 인스턴스 유형이 지원되는지 여부를 나타냅니다.

```
aws ec2 describe-instance-type-offerings \
    --filters Name=instance-type,Values=c5.xlarge \
    --region us-east-2
```

다음 `describe-instance-type-offerings` 예제에서는 지정된 리전에서 지원되는 모든 C5 인스턴스 유형을 나열합니다.

```
aws ec2 describe-instance-type-offerings \
    --filters Name=instance-type,Values=c5* \
    --query "InstanceTypeOfferings[].InstanceType" \
    --region us-east-2
```

출력:

```
[
    "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"
]
```

- 자세한 API 내용은 명령 참조DescribeInstanceTypeOfferings의 섹션을 참조하세요. AWS CLI

### describe-instance-types

다음 코드 예시에서는 describe-instance-types을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 인스턴스 유형을 설명하는 방법

다음 describe-instance-types 예제에서는 지정된 인스턴스 유형의 세부 정보를 표시합니다.

```
aws ec2 describe-instance-types \
    --instance-types t2.micro
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 인스턴스 유형을 참조하세요.

예제 2: 사용 가능한 인스턴스 유형을 필터링하는 방법

필터를 지정하여 결과 범위를 특정 특성의 인스턴스 유형으로 지정할 수 있습니다. 다음 `describe-instance-types` 예제에서는 최대 절전 모드를 지원하는 인스턴스 유형을 나열합니다.

```
aws ec2 describe-instance-types \
    --filters Name=hibernation-supported,Values=true --query
 'InstanceTypes[*].InstanceType'
```

출력:

```
[
    "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"
]
```

자세한 내용은 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 인스턴스 유형을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeInstanceTypes의 섹션을 참조하세요. AWS CLI

### `describe-instances`

다음 코드 예시에서는 `describe-instances`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 인스턴스를 설명하는 방법

다음 `describe-instances` 예제에서는 지정된 인스턴스를 설명합니다.

```
aws ec2 describe-instances \
    --instance-ids i-1234567890abcdef0
```

출력:

```
{
    "Reservations": [
        {
            "Groups": [],
            "Instances": [
                {
                    "AmiLaunchIndex": 0,
                    "ImageId": "ami-0abcdef1234567890",
                    "InstanceId": "i-1234567890abcdef0",
                    "InstanceType": "t3.nano",
                    "KeyName": "my-key-pair",
                    "LaunchTime": "2022-11-15T10:48:59+00:00",
```

```
                    "Monitoring": {
                        "State": "disabled"
                    },
                    "Placement": {
                        "AvailabilityZone": "us-east-2a",
                        "GroupName": "",
                        "Tenancy": "default"
                    },
                    "PrivateDnsName": "ip-10-0-0-157.us-east-2.compute.internal",
                    "PrivateIpAddress": "10-0-0-157",
                    "ProductCodes": [],
                    "PublicDnsName": "ec2-34-253-223-13.us-
east-2.compute.amazonaws.com",
                    "PublicIpAddress": "34.253.223.13",
                    "State": {
                        "Code": 16,
                        "Name": "running"
                    },
                    "StateTransitionReason": "",
                    "SubnetId": "subnet-04a636d18e83cfacb",
                    "VpcId": "vpc-1234567890abcdef0",
                    "Architecture": "x86_64",
                    "BlockDeviceMappings": [
                        {
                            "DeviceName": "/dev/xvda",
                            "Ebs": {
                                "AttachTime": "2022-11-15T10:49:00+00:00",
                                "DeleteOnTermination": true,
                                "Status": "attached",
                                "VolumeId": "vol-02e6ccdca7de29cf2"
                            }
                        }
                    ],
                    "ClientToken": "1234abcd-1234-abcd-1234-d46a8903e9bc",
                    "EbsOptimized": true,
                    "EnaSupport": true,
                    "Hypervisor": "xen",
                    "IamInstanceProfile": {
                        "Arn": "arn:aws:iam::111111111111:instance-profile/
AmazonSSMRoleForInstancesQuickSetup",
                        "Id": "11111111111111111111"
                    },
                    "NetworkInterfaces": [
                        {
```

```
                            "Association": {
                                "IpOwnerId": "amazon",
                                "PublicDnsName": "ec2-34-253-223-13.us-
east-2.compute.amazonaws.com",
                                "PublicIp": "34.253.223.13"
                            },
                            "Attachment": {
                                "AttachTime": "2022-11-15T10:48:59+00:00",
                                "AttachmentId": "eni-attach-1234567890abcdefg",
                                "DeleteOnTermination": true,
                                "DeviceIndex": 0,
                                "Status": "attached",
                                "NetworkCardIndex": 0
                            },
                            "Description": "",
                            "Groups": [
                                {
                                    "GroupName": "launch-wizard-146",
                                    "GroupId": "sg-1234567890abcdefg"
                                }
                            ],
                            "Ipv6Addresses": [],
                            "MacAddress": "00:11:22:33:44:55",
                            "NetworkInterfaceId": "eni-1234567890abcdefg",
                            "OwnerId": "104024344472",
                            "PrivateDnsName": "ip-10-0-0-157.us-
east-2.compute.internal",
                            "PrivateIpAddress": "10-0-0-157",
                            "PrivateIpAddresses": [
                                {
                                    "Association": {
                                        "IpOwnerId": "amazon",
                                        "PublicDnsName": "ec2-34-253-223-13.us-
east-2.compute.amazonaws.com",
                                        "PublicIp": "34.253.223.13"
                                    },
                                    "Primary": true,
                                    "PrivateDnsName": "ip-10-0-0-157.us-
east-2.compute.internal",
                                    "PrivateIpAddress": "10-0-0-157"
                                }
                            ],
                            "SourceDestCheck": true,
                            "Status": "in-use",
```

```
            "SubnetId": "subnet-1234567890abcdefg",
            "VpcId": "vpc-1234567890abcdefg",
            "InterfaceType": "interface"
        }
    ],
    "RootDeviceName": "/dev/xvda",
    "RootDeviceType": "ebs",
    "SecurityGroups": [
        {
            "GroupName": "launch-wizard-146",
            "GroupId": "sg-1234567890abcdefg"
        }
    ],
    "SourceDestCheck": true,
    "Tags": [
        {
            "Key": "Name",
            "Value": "my-instance"
        }
    ],
    "VirtualizationType": "hvm",
    "CpuOptions": {
        "CoreCount": 1,
        "ThreadsPerCore": 2
    },
    "CapacityReservationSpecification": {
        "CapacityReservationPreference": "open"
    },
    "HibernationOptions": {
        "Configured": false
    },
    "MetadataOptions": {
        "State": "applied",
        "HttpTokens": "optional",
        "HttpPutResponseHopLimit": 1,
        "HttpEndpoint": "enabled",
        "HttpProtocolIpv6": "disabled",
        "InstanceMetadataTags": "enabled"
    },
    "EnclaveOptions": {
        "Enabled": false
    },
    "PlatformDetails": "Linux/UNIX",
    "UsageOperation": "RunInstances",
```

```
                    "UsageOperationUpdateTime": "2022-11-15T10:48:59+00:00",
                    "PrivateDnsNameOptions": {
                        "HostnameType": "ip-name",
                        "EnableResourceNameDnsARecord": true,
                        "EnableResourceNameDnsAAAARecord": false
                    },
                    "MaintenanceOptions": {
                        "AutoRecovery": "default"
                    }
                }
            ],
            "OwnerId": "111111111111",
            "ReservationId": "r-1234567890abcdefg"
        }
    ]
}
```

예제 2: 지정된 유형으로 인스턴스를 필터링하는 방법

다음 `describe-instances` 예제에서는 필터를 사용하여 결과 범위를 지정된 유형의 인스턴스로 지정합니다.

```
aws ec2 describe-instances \
    --filters Name=instance-type,Values=m5.large
```

예제 출력은 예제 1을 참조하세요.

자세한 내용은 Amazon EC2 사용 설명서의 를 사용하여 목록 및 필터를 CLI 참조하세요.

예제 3: 지정된 유형 및 가용 영역으로 인스턴스를 필터링하는 방법

다음 `describe-instances` 예제에서는 여러 필터를 사용하여 결과 범위를 지정된 가용 영역에도 있는 지정된 유형의 인스턴스로 지정합니다.

```
aws ec2 describe-instances \
    --filters Name=instance-type,Values=t2.micro,t3.micro Name=availability-
zone,Values=us-east-2c
```

예제 출력은 예제 1을 참조하세요.

예제 4: JSON 파일을 사용하여 지정된 유형 및 가용 영역이 있는 인스턴스를 필터링하려면

다음 describe-instances 예제에서는 JSON 입력 파일을 사용하여 이전 예제와 동일한 필터링을 수행합니다. 필터가 더 복잡해지면 JSON 파일에서 더 쉽게 지정할 수 있습니다.

```
aws ec2 describe-instances \
    --filters file://filters.json
```

filters.json의 콘텐츠:

```
[
    {
        "Name": "instance-type",
        "Values": ["t2.micro", "t3.micro"]
    },
    {
        "Name": "availability-zone",
        "Values": ["us-east-2c"]
    }
]
```

예제 출력은 예제 1을 참조하세요.

예제 5: 지정된 소유자 태그로 인스턴스를 필터링하는 방법

다음 describe-instances 예제에서는 태그 필터를 사용하여 결과 범위를 태그 값에 관계없이 지정된 태그 키(소유자)의 태그가 있는 인스턴스로 지정합니다.

```
aws ec2 describe-instances \
    --filters "Name=tag-key,Values=Owner"
```

예제 출력은 예제 1을 참조하세요.

예제 6: 지정된 my-team 태그 값으로 인스턴스를 필터링하는 방법

다음 describe-instances 예제에서는 태그 필터를 사용하여 결과 범위를 태그 값에 관계없이 지정된 태그 값(my-team)의 태그가 있는 인스턴스로 지정합니다.

```
aws ec2 describe-instances \
    --filters "Name=tag-value,Values=my-team"
```

예제 출력은 예제 1을 참조하세요.

예제 7: 지정된 소유자 태그와 my-team 값으로 인스턴스를 필터링하는 방법

다음 `describe-instances` 예제에서는 태그 필터를 사용하여 결과 범위를 지정된 태그의 인스턴스(소유자=my-team)로 지정합니다.

```
aws ec2 describe-instances \
    --filters "Name=tag:Owner,Values=my-team"
```

예제 출력은 예제 1을 참조하세요.

예제 8: 모든 인스턴스IDs에 대해 인스턴스 및 서브넷만 표시하려면

다음 `describe-instances` 예제에서는 `--query` 파라미터를 사용하여 IDs 모든 인스턴스의 인스턴스와 서브넷만 JSON 형식으로 표시합니다.

Linux 및 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
```

출력:

```
[
    {
        "Instance": "i-057750d42936e468a",
        "Subnet": "subnet-069beee9b12030077"
    },
    {
        "Instance": "i-001efd250faaa6ffa",
        "Subnet": "subnet-0b715c6b7db68927a"
    },
    {
        "Instance": "i-027552a73f021f3bd",
        "Subnet": "subnet-0250c25a1f4e15235"
    }
```

```
    ...
]
```

예제 9: 지정된 유형의 인스턴스를 필터링하고 해당 인스턴스만 표시하려면 IDs

다음 `describe-instances` 예제에서는 필터를 사용하여 결과를 지정된 유형의 인스턴스로 범위를 지정하고 `--query` 파라미터를 사용하여 인스턴스만 표시합니다IDs.

```
aws ec2 describe-instances \
    --filters "Name=instance-type,Values=t2.micro" \
    --query "Reservations[*].Instances[*].[InstanceId]" \
    --output text
```

출력:

```
i-031c0dc19de2fb70c
i-00d8bff789a736b75
i-0b715c6b7db68927a
i-0626d4edd54f1286d
i-00b8ae04f9f99908e
i-0fc71c25d2374130c
```

예제 10: 지정된 유형의 인스턴스를 필터링하고 인스턴스IDs, 가용 영역 및 지정된 태그 값만 표시하려면

다음 `describe-instances` 예제에서는 이름이 `tag-key`인 태그의 인스턴스에 대해 인스턴스 ID, 가용 영역, `Name` 태그 값을 테이블 형식으로 표시합니다.

Linux 및 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
```

출력:

```
-------------------------------------------------------------
|                     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     |
+--------------+-----------------------+--------------------+
```

예제 11: 파티션 배치 그룹에서 인스턴스를 설명하는 방법

다음 `describe-instances` 예제에서는 지정된 인스턴스를 설명합니다. 응답에는 인스턴스의 배치 정보가 포함되며, 이 정보는 인스턴스의 배치 그룹 이름 및 파티션 번호를 포함합니다.

```
aws ec2 describe-instances \
    --instance-ids i-0123a456700123456 \
    --query "Reservations[*].Instances[*].Placement"
```

출력:

```
[
    [
        {
            "AvailabilityZone": "us-east-1c",
            "GroupName": "HDFS-Group-A",
            "PartitionNumber": 3,
            "Tenancy": "default"
        }

    ]
]
```

자세한 내용은 Amazon EC2 사용 설명서의 배치 그룹의 인스턴스 설명을 참조하세요.

예제 12: 지정된 배치 그룹과 파티션 번호로 인스턴스를 필터링하는 방법

다음 `describe-instances` 예제에서는 결과를 지정된 배치 그룹 및 파티션 번호의 인스턴스로만 필터링합니다.

```
aws ec2 describe-instances \
    --filters "Name=placement-group-name,Values=HDFS-Group-A" "Name=placement-partition-number,Values=7"
```

다음에서는 출력의 관련 정보만 보여줍니다.

```
"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"
        }
    ],
```

자세한 내용은 Amazon EC2 사용 설명서의 배치 그룹의 인스턴스 설명을 참조하세요.

예제 13: 인스턴스 메타데이터에서 태그에 액세스할 수 있도록 구성된 인스턴스를 필터링하는 방법

다음 `describe-instances` 예제에서는 인스턴스 메타데이터에서 인스턴스 태그에 액세스할 수 있도록 구성된 인스턴스로만 결과를 필터링합니다.

```
aws ec2 describe-instances \
    --filters "Name=metadata-options.instance-metadata-tags,Values=enabled" \
```

```
    --query "Reservations[*].Instances[*].InstanceId" \
    --output text
```

다음에서는 예상 출력을 보여줍니다.

```
i-1234567890abcdefg
i-abcdefg1234567890
i-11111111aaaaaaaaa
i-aaaaaaaa111111111
```

자세한 내용은 Amazon EC2 사용 설명서의 인스턴스 메타데이터에서 인스턴스 태그 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeInstances의 섹션을 참조하세요. AWS CLI

### describe-internet-gateways

다음 코드 예시에서는 describe-internet-gateways을 사용하는 방법을 보여 줍니다.

AWS CLI

인터넷 게이트웨이를 설명하려면

다음 describe-internet-gateways 예제에서는 지정된 인터넷 게이트웨이를 설명합니다.

```
aws ec2 describe-internet-gateways \
    --internet-gateway-ids igw-0d0fb496b3EXAMPLE
```

출력:

```
{
    "InternetGateways": [
        {
            "Attachments": [
                {
                    "State": "available",
                    "VpcId": "vpc-0a60eb65b4EXAMPLE"
                }
            ],
            "InternetGatewayId": "igw-0d0fb496b3EXAMPLE",
            "OwnerId": "123456789012",
            "Tags": [
```

```
                {
                    "Key": "Name",
                    "Value": "my-igw"
                }
            ]
        }
    ]
}
```

자세한 내용은 Amazon VPC 사용 설명서의 인터넷 게이트웨이를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeInternetGateways의 섹션을 참조하세요. AWS CLI

### describe-ipam-pools

다음 코드 예시에서는 describe-ipam-pools을 사용하는 방법을 보여 줍니다.

AWS CLI

IPAM 풀에 대한 세부 정보를 보려면

다음 describe-ipam-pools 예제에서는 풀에 대한 세부 정보를 보여줍니다.

(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
```

출력:

```
{
    "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"
                }
            ]
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeIpamPools의 섹션을 참조하세요. AWS CLI

### describe-ipam-resource-discoveries

다음 코드 예시에서는 describe-ipam-resource-discoveries을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 리소스 검색의 전체 세부 정보 보기

이 예제에서는 관리자가 조직 IPAM 내 리소스의 IP 주소를 관리하고 모니터링할 수 있도록 리소스 검색을 생성하고 다른 AWS 조직의 관리자와 공유IPAM하려는 위임된 관리자입니다.

이 예제는 다음과 같은 경우에 유용할 수 있습니다.

리소스 검색을 생성하려고 했지만 한도 1에 도달하는 오류가 발생했습니다. 리소스 검색을 이미 생성하여 계정에서 보려는 것 같습니다. 리전에 에서 검색하지 않는 리소스가 있습니다IPAM. 리소스에 대해 `--operating-regions` 정의된 를 보고 적절한 리전을 운영 리전으로 추가하여 해당 리전의 리소스를 검색할 수 있도록 하려고 합니다.

다음 `describe-ipam-resource-discoveries` 예제에서는 AWS 계정의 리소스 검색에 대한 세부 정보를 나열합니다. AWS 리전당 하나의 리소스 검색을 수행할 수 있습니다.

```
aws ec2 describe-ipam-resource-discoveries \
    --region us-east-1
```

출력:

```
{
    "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": []
    }
]
}
```

자세한 내용은 Amazon VPC IPAM 사용 설명서의 조직 외부 계정IPAM과 통합을 참조하세요.

예제 2: 리소스 검색만 보기 IDs

다음 `describe-ipam-resource-discoveries` 예제에서는 AWS 계정의 리소스 검색 ID를 나열합니다. AWS 리전당 하나의 리소스 검색을 수행할 수 있습니다.

```
aws ec2 describe-ipam-resource-discoveries \
    --query "IpamResourceDiscoveries[*].IpamResourceDiscoveryId" \
    --output text
```

출력:

```
ipam-res-disco-0481e39b242860333
```

자세한 내용은 Amazon VPC IPAM 사용 설명서의 조직 외부 계정IPAM과 통합을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeIpamResourceDiscoveries의 섹션을 참조하세요. AWS CLI

### describe-ipam-resource-discovery-associations

다음 코드 예시에서는 describe-ipam-resource-discovery-associations을 사용하는 방법을 보여 줍니다.

AWS CLI

와의 모든 리소스 검색 연결을 보려면 IPAM

이 예제에서는 와 리소스 검색을 연결하여 다른 계정을 와 IPAM 통합하는 IPAM 위임된 관리자입니다IPAM. IPAM 에서 리소스 검색의 운영 리전에서 리소스를 예상대로 검색하지 않는 것을 알았습니다. 리소스 검색의 상태 및 상태를 확인하여 리소스 검색을 생성한 계정이 여전히 활성 상태이고 리소스 검색이 여전히 공유 중인지 확인해야 합니다.

는 의 홈 리전이어야 --region 합니다IPAM.

다음 describe-ipam-resource-discovery-associations 예제에서는 AWS 계정의 리소스 검색 연결을 나열합니다.

```
aws ec2 describe-ipam-resource-discovery-associations \
    --region us-east-1
```

출력:

```
{
    "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": []
        }
    ]
}
```

이 예제에서는 이 명령을 실행한 후 기본값이 아닌 리소스 검색("IsDefault": false ``) that is ``"ResourceDiscoveryStatus": "not-found" 및 )이 하나 있음을 발견합니다"State": "create-complete". 리소스 검색 소유자의 계정이 닫혔습니다. 또 다른 경우 가 "ResourceDiscoveryStatus": "not-found" 및 인 것을 발견하면 다음 중 하나가 발생했음을 "State": "associate-complete"나타냅니다.

리소스 검색 소유자가 리소스 검색을 삭제했습니다. 리소스 검색 소유자는 리소스 검색을 공유하지 않았습니다.

자세한 내용은 Amazon VPC IPAM 사용 설명서의 조직 외부 계정IPAM과 통합을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeIpamResourceDiscoveryAssociations의 섹션을 참조하세요. AWS CLI

### `describe-ipam-scopes`

다음 코드 예시에서는 `describe-ipam-scopes`을 사용하는 방법을 보여 줍니다.

AWS CLI

IPAM 범위에 대한 세부 정보를 보려면

다음 `describe-ipam-scopes` 예제에서는 범위에 대한 세부 정보를 보여줍니다.

```
aws ec2 describe-ipam-scopes \
    --filters Name=owner-id,Values=123456789012 Name=ipam-
id,Values=ipam-08440e7a3acde3908
```

출력:

```
{
    "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"
                }
            ]
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeIpamScopes의 섹션을 참조하세요. AWS CLI

### `describe-ipams`

다음 코드 예시에서는 `describe-ipams`을 사용하는 방법을 보여 줍니다.

AWS CLI

에 대한 세부 정보를 보려면 IPAM

다음 `describe-ipams` 예제에서는 의 세부 정보를 보여줍니다IPAM.

```
aws ec2 describe-ipams \
    --filters Name=owner-id,Values=123456789012
```

출력:

```
{
    "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"
                }
            ]
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeIpams의 섹션을 참조하세요. AWS CLI

### `describe-ipv6-pools`

다음 코드 예시에서는 `describe-ipv6-pools`을 사용하는 방법을 보여 줍니다.

AWS CLI

IPv6 주소 풀을 설명하려면

다음 `describe-ipv6-pools` 예제에서는 모든 IPv6 주소 풀에 대한 세부 정보를 표시합니다.

```
aws ec2 describe-ipv6-pools
```

출력:

```
{
    "Ipv6Pools": [
        {
            "PoolId": "ipv6pool-ec2-012345abc12345abc",
            "PoolCidrBlocks": [
                {
                    "Cidr": "2001:db8:123::/48"
                }
            ],
            "Tags": [
                {
                    "Key": "pool-1",
                    "Value": "public"
                }
            ]
        }
    ]
}
```

- API 자세한 내용은 AWS CLI 명령 참조의 DescribeIpv6Pools을 참조하세요.

### describe-key-pairs

다음 코드 예시에서는 describe-key-pairs을 사용하는 방법을 보여 줍니다.

AWS CLI

키 페어를 표시하는 방법

다음 describe-key-pairs 예제는 지정된 키 페어에 대한 정보를 표시합니다.

```
aws ec2 describe-key-pairs \
    --key-names my-key-pair
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon EC2 사용 설명서의 퍼블릭 키 설명을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeKeyPairs의 섹션을 참조하세요. AWS CLI

### describe-launch-template-versions

다음 코드 예시에서는 describe-launch-template-versions을 사용하는 방법을 보여 줍니다.

AWS CLI

시작 템플릿 버전을 설명하려면

이 예제에서는 지정된 시작 템플릿의 버전을 설명합니다.

명령:

```
aws ec2 describe-launch-template-versions --launch-template-id lt-068f72b72934aff71
```

출력:

```
{
  "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeLaunchTemplateVersions의 섹션을 참조하세요. AWS CLI

### `describe-launch-templates`

다음 코드 예시에서는 `describe-launch-templates`을 사용하는 방법을 보여 줍니다.

AWS CLI

시작 템플릿을 설명하려면

이 예제에서는 시작 템플릿을 설명합니다.

명령:

```
aws ec2 describe-launch-templates
```

출력:

```
{
    "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"
        },
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeLaunchTemplates의 섹션을 참조하세요. AWS CLI

### describe-local-gateway-route-table-virtual-interface-group-associations

다음 코드 예시에서는 describe-local-gateway-route-table-virtual-interface-group-associations을 사용하는 방법을 보여 줍니다.

AWS CLI

가상 인터페이스 그룹과 로컬 게이트웨이 라우팅 테이블 간의 연결을 설명하려면

다음 describe-local-gateway-route-table-virtual-interface-group-associations 예제에서는 AWS 계정의 가상 인터페이스 그룹과 로컬 게이트웨이 라우팅 테이블 간의 연결을 설명합니다.

```
aws ec2 describe-local-gateway-route-table-virtual-interface-group-associations
```

출력:

```
{
    "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": []
        }
    ]
}
```

자세한 내용은 Outposts 사용 설명서의 로컬 게이트웨이 작업을 참조하세요. AWS

- 자세한 API 내용은 명령 참조DescribeLocalGatewayRouteTableVirtualInterfaceGroupAssociations의 섹션을 참조하세요. AWS CLI

### describe-local-gateway-route-table-vpc-associations

다음 코드 예시에서는 describe-local-gateway-route-table-vpc-associations을 사용하는 방법을 보여 줍니다.

AWS CLI

VPCs 및 로컬 게이트웨이 라우팅 테이블 간의 연결을 설명하려면

다음 describe-local-gateway-route-table-vpc-associations 예제에서는 VPCs 및 로컬 게이트웨이 라우팅 테이블 간의 지정된 연결에 대한 정보를 표시합니다.

```
aws ec2 describe-local-gateway-route-table-vpc-associations \
    --local-gateway-route-table-vpc-association-ids lgw-vpc-assoc-0e0f27af15EXAMPLE
```

출력:

```
{
    "LocalGatewayRouteTableVpcAssociation": {
        "LocalGatewayRouteTableVpcAssociationId": "lgw-vpc-assoc-0e0f27af1EXAMPLE",
        "LocalGatewayRouteTableId": "lgw-rtb-059615ef7dEXAMPLE",
        "LocalGatewayId": "lgw-09b493aa7cEXAMPLE",
        "VpcId": "vpc-0efe9bde08EXAMPLE",
        "State": "associated"
    }
}
```

자세한 내용은  Outposts 사용 설명서의 로컬 게이트웨이 라우팅 테이블을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeLocalGatewayRouteTableVpcAssociations의 섹션을 참조하세요. AWS CLI

### describe-local-gateway-route-tables

다음 코드 예시에서는 describe-local-gateway-route-tables을 사용하는 방법을 보여 줍니다.

AWS CLI

로컬 게이트웨이 라우팅 테이블을 설명하려면

다음 describe-local-gateway-route-tables 예제에서는 로컬 게이트웨이 라우팅 테이블에 대한 세부 정보를 보여줍니다.

```
aws ec2 describe-local-gateway-route-tables
```

출력:

```
{
    "LocalGatewayRouteTables": [
        {
            "LocalGatewayRouteTableId": "lgw-rtb-059615ef7deEXAMPLE",
            "LocalGatewayId": "lgw-09b493aa7cEXAMPLE",
```

```
            "OutpostArn": "arn:aws:outposts:us-west-2:111122223333:outpost/
op-0dc11b66edEXAMPLE",
            "State": "available"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeLocalGatewayRouteTables의 섹션을 참조하세요. AWS CLI

### `describe-local-gateway-virtual-interface-groups`

다음 코드 예시에서는 `describe-local-gateway-virtual-interface-groups`을 사용하는 방법을 보여 줍니다.

AWS CLI

로컬 게이트웨이 가상 인터페이스 그룹을 설명하려면

다음 `describe-local-gateway-virtual-interface-groups` 예제에서는 AWS 계정의 로컬 게이트웨이 가상 인터페이스 그룹에 대해 설명합니다.

```
aws ec2 describe-local-gateway-virtual-interface-groups
```

출력:

```
{
    "LocalGatewayVirtualInterfaceGroups": [
        {
            "LocalGatewayVirtualInterfaceGroupId": "lgw-vif-grp-07145b276bEXAMPLE",
            "LocalGatewayVirtualInterfaceIds": [
                "lgw-vif-01a23bc4d5EXAMPLE",
                "lgw-vif-543ab21012EXAMPLE"
            ],
            "LocalGatewayId": "lgw-0ab1c23d4eEXAMPLE",
            "OwnerId": "123456789012",
            "Tags": []
        }
    ]
}
```

자세한 내용은 Outposts 사용 설명서의 로컬 게이트웨이 작업을 참조하세요. AWS

- 자세한 API 내용은 명령 참조DescribeLocalGatewayVirtualInterfaceGroups의 섹션을 참조하세요. AWS CLI

### describe-local-gateway-virtual-interfaces

다음 코드 예시에서는 describe-local-gateway-virtual-interfaces을 사용하는 방법을 보여 줍니다.

AWS CLI

로컬 게이트웨이 가상 인터페이스를 설명하려면

다음 describe-local-gateway-virtual-interfaces 예제에서는 AWS 계정의 로컬 게이트웨이 가상 인터페이스를 설명합니다.

```
aws ec2 describe-local-gateway-virtual-interfaces
```

출력:


```
{
    "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": []
```

```
        }
    ]
}
```

자세한 내용은 Outposts 사용 설명서의 로컬 게이트웨이 작업을 참조하세요. AWS

- 자세한 API 내용은 명령 참조DescribeLocalGatewayVirtualInterfaces의 섹션을 참조하세요. AWS CLI

### `describe-local-gateways`

다음 코드 예시에서는 `describe-local-gateways`을 사용하는 방법을 보여 줍니다.

AWS CLI

로컬 게이트웨이를 설명하려면

다음 `describe-local-gateways` 예제에서는 사용 가능한 로컬 게이트웨이에 대한 세부 정보를 표시합니다.

```
aws ec2 describe-local-gateways
```

출력:

```
{
    "LocalGateways": [
        {
            "LocalGatewayId": "lgw-09b493aa7cEXAMPLE",
            "OutpostArn": "arn:aws:outposts:us-west-2:123456789012:outpost/
op-0dc11b66ed59f995a",
            "OwnerId": "123456789012",
            "State": "available"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeLocalGateways의 섹션을 참조하세요. AWS CLI

### `describe-locked-snapshots`

다음 코드 예시에서는 `describe-locked-snapshots`을 사용하는 방법을 보여 줍니다.

AWS CLI

스냅샷의 잠금 상태를 설명하려면

다음 `describe-locked-snapshots` 예제에서는 지정된 스냅샷의 잠금 상태를 설명합니다.

```
aws ec2 describe-locked-snapshots \
    --snapshot-ids snap-0b5e733b4a8df6e0d
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon EBS 사용 설명서의 스냅샷 잠금을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeLockedSnapshots의 섹션을 참조하세요. AWS CLI

### describe-managed-prefix-lists

다음 코드 예시에서는 `describe-managed-prefix-lists`을 사용하는 방법을 보여 줍니다.

AWS CLI

관리형 접두사 목록을 설명하려면

다음 `describe-managed-prefix-lists` 예제에서는 AWS 계정 에서 소유한 접두사 목록을 설명합니다123456789012.

```
aws ec2 describe-managed-prefix-lists \
    --filters Name=owner-id,Values=123456789012
```

출력:

```
{
    "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"
      }
  ]
}
```

자세한 내용은 Amazon VPC 사용 설명서의 관리형 접두사 목록을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeManagedPrefixLists의 섹션을 참조하세요. AWS CLI

### `describe-moving-addresses`

다음 코드 예시에서는 `describe-moving-addresses`을 사용하는 방법을 보여 줍니다.

AWS CLI

이동하는 주소를 설명하려면

이 예제에서는 이동하는 모든 탄력적 IP 주소를 설명합니다.

명령:

```
aws ec2 describe-moving-addresses
```

출력:

```
{
  "MovingAddressStatuses": [
    {
      "PublicIp": "198.51.100.0",
      "MoveStatus": "MovingToVpc"
    }
  ]
}
```

이 예제에서는 EC2-VPC 플랫폼으로 이동하는 모든 주소를 설명합니다.

명령:

```
aws ec2 describe-moving-addresses --filters Name=moving-status,Values=MovingToVpc
```

- 자세한 API 내용은 명령 참조DescribeMovingAddresses의 섹션을 참조하세요. AWS CLI

### describe-nat-gateways

다음 코드 예시에서는 describe-nat-gateways을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 퍼블릭 NAT 게이트웨이 설명

다음 describe-nat-gateways 예제에서는 지정된 퍼블릭 NAT 게이트웨이를 설명합니다.

```
aws ec2 describe-nat-gateways \
    --nat-gateway-id nat-01234567890abcdef
```

출력:

```
{
    "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"
        }
    ]
}
```

예제 2: 프라이빗 NAT 게이트웨이 설명

다음 `describe-nat-gateways` 예제에서는 지정된 프라이빗 NAT 게이트웨이를 설명합니다.

```
aws ec2 describe-nat-gateways \
    --nat-gateway-id nat-1234567890abcdef0
```

출력:


```
{
    "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"
        }
    ]
}
```


자세한 내용은 Amazon VPC 사용 설명서의 NAT 게이트웨이를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeNatGateways의 섹션을 참조하세요. AWS CLI

### describe-network-acls

다음 코드 예시에서는 describe-network-acls을 사용하는 방법을 보여 줍니다.

AWS CLI

네트워크를 설명하려면 ACLs

다음 describe-network-acls 예제에서는 네트워크 에 대한 세부 정보를 검색합니다ACLs.

```
aws ec2 describe-network-acls
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS VPC 사용 설명서의 네트워크를 ACLs 참조하세요.

- 자세한 API 내용은 명령 참조DescribeNetworkAcls의 섹션을 참조하세요. AWS CLI

### describe-network-insights-access-scope-analyses

다음 코드 예시에서는 describe-network-insights-access-scope-analyses을 사용하는 방법을 보여 줍니다.

AWS CLI

Network Insights 액세스 범위 분석을 설명하려면

다음 describe-network-insights-access-scope-analyses 예제에서는 AWS 계정의 액세스 범위 분석을 설명합니다.

```
aws ec2 describe-network-insights-access-scope-analyses \
    --region us-east-1
```

출력:

```
{
    "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": []
        }
    ]
}
```

자세한 내용은 Network Access Analyzer 가이드의 를 사용하여 AWS CLI Network Access Analyzer 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeNetworkInsightsAccessScopeAnalyses의 섹션을 참조하세요. AWS CLI

### describe-network-insights-access-scopes

다음 코드 예시에서는 describe-network-insights-access-scopes을 사용하는 방법을 보여 줍니다.

AWS CLI

Network Insights 액세스 범위를 설명하려면

다음 `describe-network-insights-access-scopes` 예제에서는 AWS 계정의 액세스 범위 분석을 설명합니다.

```
aws ec2 describe-network-insights-access-scopes \
    --region us-east-1
```

출력:

```
{
    "NetworkInsightsAccessScopes": [
        {
            "NetworkInsightsAccessScopeId": "nis-123456789111",
            "NetworkInsightsAccessScopeArn": "arn:aws:ec2:us-
east-1:123456789012:network-insights-access-scope/nis-123456789111",
            "CreatedDate": "2021-11-29T21:12:41.416000+00:00",
            "UpdatedDate": "2021-11-29T21:12:41.416000+00:00",
            "Tags": []
        }
    ]
}
```

자세한 내용은 Network Access Analyzer 안내서의 를 사용하여 AWS CLI Network Access Analyzer 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeNetworkInsightsAccessScopes의 섹션을 참조하세요. AWS CLI

### `describe-network-insights-analyses`

다음 코드 예시에서는 `describe-network-insights-analyses`을 사용하는 방법을 보여 줍니다.

AWS CLI

경로 분석 결과를 보려면

다음 `describe-network-insights-analyses` 예제에서는 지정된 분석을 설명합니다. 이 예제에서 소스는 인터넷 게이트웨이이고 대상은 EC2 인스턴스이며 프로토콜은 입니다TCP. 분석에 성공succeeded했고(`Status` ) 경로에 도달할 수 없습니다(`NetworkPathFound` ). `false` 설

명 코드는 인스턴스의 보안 그룹에 대상 포트의 트래픽을 허용하는 규칙이 포함되어 있지 않음을 ENI_SG_RULES_MISMATCH 나타냅니다.

```
aws ec2 describe-network-insights-analyses \
    --network-insights-analysis-ids nia-02207aa13eb480c7a
```

출력:

```
{
    "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": []
        }
    ]
}
```

자세한 내용은 Reachability Analyzer 가이드의 시작하기 AWS CLI를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeNetworkInsightsAnalyses의 섹션을 참조하세요. AWS CLI

### describe-network-insights-paths

다음 코드 예시에서는 describe-network-insights-paths을 사용하는 방법을 보여 줍니다.

AWS CLI

경로를 설명하려면

다음 describe-network-insights-paths 예제에서는 지정된 경로를 설명합니다.

```
aws ec2 describe-network-insights-paths \
    --network-insights-path-ids nip-0b26f224f1d131fa8
```

출력:

```
{
    "NetworkInsightsPaths": [
        {
            "NetworkInsightsPathId": "nip-0b26f224f1d131fa8",
            "NetworkInsightsPathArn": "arn:aws:ec2:us-east-1:123456789012:network-
insights-path/nip-0b26f224f1d131fa8",
            "CreatedDate": "2021-01-20T22:43:46.933Z",
            "Source": "igw-0797cccdc9d73b0e5",
            "Destination": "i-0495d385ad28331c7",
            "Protocol": "tcp"
        }
    ]
}
```

자세한 내용은 Reachability Analyzer 가이드의 시작하기 AWS CLI를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeNetworkInsightsPaths의 섹션을 참조하세요. AWS CLI

### `describe-network-interface-attribute`

다음 코드 예시에서는 `describe-network-interface-attribute`을 사용하는 방법을 보여 줍니다.

AWS CLI

네트워크 인터페이스의 연결 속성을 설명하려면

이 예제 명령은 지정된 네트워크 인터페이스의 `attachment` 속성을 설명합니다.

명령:

```
aws ec2 describe-network-interface-attribute --network-interface-id eni-686ea200 --
attribute attachment
```

출력:

```
{
  "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"
  }
}
```

네트워크 인터페이스의 설명 속성을 설명하려면

이 예제 명령은 지정된 네트워크 인터페이스의 `description` 속성을 설명합니다.

명령:

```
aws ec2 describe-network-interface-attribute --network-interface-id eni-686ea200 --
attribute description
```

출력:

```
{
  "NetworkInterfaceId": "eni-686ea200",
  "Description": {
      "Value": "My description"
  }
}
```

네트워크 인터페이스의 groupSet 속성을 설명하려면

이 예제 명령은 지정된 네트워크 인터페이스의 `groupSet` 속성을 설명합니다.

명령:

```
aws ec2 describe-network-interface-attribute --network-interface-id eni-686ea200 --
attribute groupSet
```

출력:

```
{
  "NetworkInterfaceId": "eni-686ea200",
  "Groups": [
      {
          "GroupName": "my-security-group",
          "GroupId": "sg-903004f8"
      }
  ]
}
```

네트워크 인터페이스의 sourceDestCheck 속성을 설명하려면

이 예제 명령은 지정된 네트워크 인터페이스의 `sourceDestCheck` 속성을 설명합니다.

명령:

```
aws ec2 describe-network-interface-attribute --network-interface-id eni-686ea200 --
attribute sourceDestCheck
```

출력:

```
{
```

```
  "NetworkInterfaceId": "eni-686ea200",
  "SourceDestCheck": {
      "Value": true
  }
}
```

- 자세한 API 내용은 명령 참조DescribeNetworkInterfaceAttribute의 섹션을 참조하세요. AWS CLI

### describe-network-interface-permissions

다음 코드 예시에서는 describe-network-interface-permissions을 사용하는 방법을 보여 줍니다.

AWS CLI

네트워크 인터페이스 권한을 설명하려면

이 예제에서는 모든 네트워크 인터페이스 권한을 설명합니다.

명령:

```
aws ec2 describe-network-interface-permissions
```

출력:

```
{
  "NetworkInterfacePermissions": [
      {
          "PermissionState": {
              "State": "GRANTED"
          },
          "NetworkInterfacePermissionId": "eni-perm-06fd19020ede149ea",
          "NetworkInterfaceId": "eni-b909511a",
          "Permission": "INSTANCE-ATTACH",
          "AwsAccountId": "123456789012"
      }
  ]
}
```

- 자세한 API 내용은 명령 참조DescribeNetworkInterfacePermissions의 섹션을 참조하세요. AWS CLI

### describe-network-interfaces

다음 코드 예시에서는 describe-network-interfaces을 사용하는 방법을 보여 줍니다.

AWS CLI

네트워크 인터페이스를 설명하려면

이 예제에서는 모든 네트워크 인터페이스를 설명합니다.

명령:

```
aws ec2 describe-network-interfaces
```

출력:

```
{
  "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"
        }
    ]
}
```

이 예제에서는 키Purpose와 값이 인 태그가 있는 네트워크 인터페이스에 대해 설명합니다Prod.

명령:

```
aws ec2 describe-network-interfaces --filters Name=tag:Purpose,Values=Prod
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeNetworkInterfaces의 섹션을 참조하세요. AWS CLI

### describe-placement-groups

다음 코드 예시에서는 describe-placement-groups을 사용하는 방법을 보여 줍니다.

AWS CLI

배치 그룹을 설명하려면

이 예제 명령은 모든 배치 그룹을 설명합니다.

명령:

```
aws ec2 describe-placement-groups
```

출력:

```
{
    "PlacementGroups": [
        {
            "GroupName": "my-cluster",
            "State": "available",
            "Strategy": "cluster"
        },
        ...
    ]
}
```

- 자세한 API 내용은 명령 참조DescribePlacementGroups의 섹션을 참조하세요. AWS CLI

### `describe-prefix-lists`

다음 코드 예시에서는 `describe-prefix-lists`을 사용하는 방법을 보여 줍니다.

AWS CLI

접두사 목록을 설명하려면

이 예제에서는 리전에 사용 가능한 모든 접두사 목록을 나열합니다.

명령:

```
aws ec2 describe-prefix-lists
```

출력:

```
{
  "PrefixLists": [
```

```
    {
      "PrefixListName": "com.amazonaws.us-east-1.s3",
      "Cidrs": [
        "54.231.0.0/17"
      ],
      "PrefixListId": "pl-63a5400a"
    }
  ]
}
```

- 자세한 API 내용은 명령 참조DescribePrefixLists의 섹션을 참조하세요. AWS CLI

### describe-principal-id-format

다음 코드 예시에서는 describe-principal-id-format을 사용하는 방법을 보여 줍니다.

AWS CLI

긴 ID 형식이 활성화된 IAM 사용자 및 역할의 ID 형식을 설명하려면

다음 describe-principal-id-format 예제에서는 루트 사용자, 모든 IAM 역할 및 긴 ID 형식이 활성화된 모든 IAM 사용자의 ID 형식을 설명합니다.

```
aws ec2 describe-principal-id-format \
    --resource instance
```

출력:

```
{
    "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
                },
            ]
        },
        ...
    ]
}
```

- 자세한 API 내용은 명령 참조DescribePrincipalIdFormat의 섹션을 참조하세요. AWS CLI

### `describe-public-ipv4-pools`

다음 코드 예시에서는 `describe-public-ipv4-pools`을 사용하는 방법을 보여 줍니다.

AWS CLI

퍼블릭 IPv4 주소 풀을 설명하려면

다음 `describe-public-ipv4-pools` 예제에서는 Bring Your Own IP Addresses()를 사용하여 퍼블릭 IPv4 주소 범위를 프로비저닝할 때 생성된 주소 풀에 대한 세부 정보를 표시합니다BYOIP.

```
aws ec2 describe-public-ipv4-pools
```

출력:

```
{
    "PublicIpv4Pools": [
        {
            "PoolId": "ipv4pool-ec2-1234567890abcdef0",
            "PoolAddressRanges": [
                {
                    "FirstAddress": "203.0.113.0",
                    "LastAddress": "203.0.113.255",
                    "AddressCount": 256,
                    "AvailableAddressCount": 256
                }
            ],
            "TotalAddressCount": 256,
```

```
            "TotalAvailableAddressCount": 256
        }
    ]
}
```

- API 자세한 내용은 AWS CLI 명령 참조의 DescribePublicIpv4Pools을 참조하세요.

### `describe-regions`

다음 코드 예시에서는 `describe-regions`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 활성화된 모든 리전을 설명하는 방법

다음 `describe-regions` 예제에서는 계정에서 활성화된 모든 리전을 설명합니다.

```
aws ec2 describe-regions
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon EC2 사용 설명서의 리전 및 영역을 참조하세요.

예제 2: 이름에 특정 문자열이 포함된 엔드포인트가 있는 활성화된 리전을 설명하는 방법

다음 `describe-regions` 예제에서는 엔드포인트에 'us' 문자열이 포함된 활성화한 모든 리전을 설명합니다.

```
aws ec2 describe-regions \
    --filters "Name=endpoint,Values=*us*"
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon EC2 사용 설명서의 리전 및 영역을 참조하세요.

예제 3: 모든 리전을 설명하는 방법

다음 `describe-regions` 예제에서는 비활성화된 리전을 포함하여 사용 가능한 모든 리전을 설명합니다.

```
aws ec2 describe-regions \
    --all-regions
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon EC2 사용 설명서의 리전 및 영역을 참조하세요.

예제 4: 리전 이름만 나열하는 방법

다음 describe-regions 예제에서는 --query 파라미터를 사용하여 출력을 필터링하고 리전 이름만 텍스트로 반환합니다.

```
aws ec2 describe-regions \
    --all-regions \
    --query "Regions[].{Name:RegionName}" \
    --output text
```

출력:

```
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
```

자세한 내용은 Amazon EC2 사용 설명서의 리전 및 영역을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeRegions의 섹션을 참조하세요. AWS CLI

### describe-replace-root-volume-tasks

다음 코드 예시에서는 describe-replace-root-volume-tasks을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 특정 루트 볼륨 교체 작업에 대한 정보를 보려면

다음 `describe-replace-root-volume-tasks` 예제에서는 루트 볼륨 교체 작업 replacevol-0111122223333abcd를 설명합니다.

```
aws ec2 describe-replace-root-volume-tasks \
    --replace-root-volume-task-ids replacevol-0111122223333abcd
```

출력:

```
{
    "ReplaceRootVolumeTasks": [
        {
            "ReplaceRootVolumeTaskId": "replacevol-0111122223333abcd",
            "Tags": [],
            "InstanceId": "i-0123456789abcdefa",
            "TaskState": "succeeded",
            "StartTime": "2022-03-14T15:16:28Z",
            "CompleteTime": "2022-03-14T15:16:52Z"
        }
    ]
}
```

자세한 내용은 Amazon Elastic Compute Cloud 사용 설명서의 루트 볼륨 교체를 참조하세요.

예제 2: 특정 인스턴스의 모든 루트 볼륨 교체 작업에 대한 정보를 보려면

다음 `describe-replace-root-volume-tasks` 예제에서는 인스턴스 i-0123456789abcdefa에 대한 모든 루트 볼륨 교체 작업을 설명합니다.

```
aws ec2 describe-replace-root-volume-tasks \
    --filters Name=instance-id,Values=i-0123456789abcdefa
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Elastic Compute Cloud 사용 설명서의 루트 볼륨 교체를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeReplaceRootVolumeTasks의 섹션을 참조하세요. AWS CLI

### describe-reserved-instances-listings

다음 코드 예시에서는 describe-reserved-instances-listings을 사용하는 방법을 보여 줍니다.

AWS CLI

예약 인스턴스 목록을 설명하려면

다음 describe-reserved-instances-listings 예제에서는 지정된 예약 인스턴스 목록에 대한 정보를 검색합니다.

```
aws ec2 describe-reserved-instances-listings \
    --reserved-instances-listing-id 5ec28771-05ff-4b9b-aa31-9e57dexample
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DescribeReservedInstancesListings의 섹션을 참조하세요. AWS CLI

### describe-reserved-instances-modifications

다음 코드 예시에서는 describe-reserved-instances-modifications을 사용하는 방법을 보여 줍니다.

AWS CLI

예약 인스턴스 수정을 설명하려면

이 예제 명령은 계정에 대해 제출된 모든 예약 인스턴스 수정 요청을 설명합니다.

명령:

```
aws ec2 describe-reserved-instances-modifications
```

출력:

```
{
    "ReservedInstancesModifications": [
        {
            "Status": "fulfilled",
            "ModificationResults": [
                {
                    "ReservedInstancesId": "93bbbca2-62f1-4d9d-b225-16bada29e6c7",
                    "TargetConfiguration": {
                        "AvailabilityZone": "us-east-1b",
                        "InstanceType": "m1.large",
                        "InstanceCount": 3
                    }
                },
                {
                     "ReservedInstancesId": "1ba8e2e3-aabb-46c3-bcf5-3fe2fda922e6",
                     "TargetConfiguration": {
                         "AvailabilityZone": "us-east-1d",
                         "InstanceType": "m1.xlarge",
                         "InstanceCount": 1
                     }
                 }
            ],
            "EffectiveDate": "2015-08-12T17:00:00.000Z",
            "CreateDate": "2015-08-12T17:52:52.630Z",
            "UpdateDate": "2015-08-12T18:08:06.698Z",
            "ClientToken": "c9adb218-3222-4889-8216-0cf0e52dc37e:
            "ReservedInstancesModificationId": "rimod-d3ed4335-b1d3-4de6-
ab31-0f13aaf46687",
            "ReservedInstancesIds": [
                {
                    "ReservedInstancesId": "b847fa93-e282-4f55-b59a-1342f5bd7c02"
                }
```

```
            ]
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeReservedInstancesModifications의 섹션을 참조하세요. AWS CLI

### `describe-reserved-instances-offerings`

다음 코드 예시에서는 `describe-reserved-instances-offerings`을 사용하는 방법을 보여 줍니다.

AWS CLI

예약 인스턴스 제공 설명

이 예제 명령은 리전에서 구매할 수 있는 모든 예약 인스턴스를 설명합니다.

명령:

```
aws ec2 describe-reserved-instances-offerings
```

출력:

```
{
  "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"
        },
    ...
}
```

옵션을 사용하여 예약 인스턴스 제안 설명

이 예제에서는 t1.micro 인스턴스 유형, Windows(Amazon VPC) 제품 및 사용률이 높은 제품 등의 사양 AWS 으로 에서 제공하는 예약 인스턴스를 나열합니다.

명령:

```
aws ec2 describe-reserved-instances-offerings --no-include-marketplace --instance-
type "t1.micro" --product-description "Windows (Amazon VPC)" --offering-type "no
 upfront"
```

출력:

```
{
    "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"
    }
]
```

```
}
```

- 자세한 API 내용은 명령 참조DescribeReservedInstancesOfferings의 섹션을 참조하세요. AWS CLI

### `describe-reserved-instances`

다음 코드 예시에서는 `describe-reserved-instances`을 사용하는 방법을 보여 줍니다.

AWS CLI

예약 인스턴스를 설명하려면

이 예제 명령은 소유한 예약 인스턴스를 설명합니다.

명령:

```
aws ec2 describe-reserved-instances
```

출력:

```
{
  "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
        },
        ...
    ]
}
```

필터를 사용하여 예약 인스턴스를 설명하려면

이 예제는 us-west-1c에 3년 t2.micro Linux/UNIX예약 인스턴스만 포함하도록 응답을 필터링합니다.

명령:

```
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
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 AWS 명령줄 인터페이스 사용 설명서의 Amazon EC2 인스턴스 사용을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeReservedInstances의 섹션을 참조하세요. AWS CLI

### `describe-route-tables`

다음 코드 예시에서는 `describe-route-tables`을 사용하는 방법을 보여 줍니다.

AWS CLI

라우팅 테이블을 설명하려면

다음 `describe-route-tables` 예제에서는 라우팅 테이블에 대한 세부 정보를 검색합니다.

```
aws ec2 describe-route-tables
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS VPC 사용 설명서의 라우팅 테이블 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeRouteTables의 섹션을 참조하세요. AWS CLI

### describe-scheduled-instance-availability

다음 코드 예시에서는 describe-scheduled-instance-availability을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 일정을 설명하려면

이 예제에서는 지정된 날짜부터 매주 일요일에 발생하는 일정을 설명합니다.

명령:

```
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
```

출력:

```
{
```

```
  "ScheduledInstanceAvailabilitySet": [
    {
        "AvailabilityZone": "us-west-2b",
        "TotalScheduledInstanceHours": 1219,
        "PurchaseToken": "eyJ2IjoiMSIsInMiOjEsImMiOi...",
        "MinTermDurationInDays": 366,
        "AvailableInstanceCount": 20,
        "Recurrence": {
            "OccurrenceDaySet": [
                1
            ],
            "Interval": 1,
            "Frequency": "Weekly",
            "OccurrenceRelativeToEnd": false
        },
        "Platform": "Linux/UNIX",
        "FirstSlotStartTime": "2016-01-31T00:00:00Z",
        "MaxTermDurationInDays": 366,
        "SlotDurationInHours": 23,
        "NetworkPlatform": "EC2-VPC",
        "InstanceType": "c4.large",
        "HourlyPrice": "0.095"
    },
    ...
  ]
}
```

결과를 좁히려면 운영 체제, 네트워크 및 인스턴스 유형을 지정하는 필터를 추가할 수 있습니다.

명령:

--filters Name=platform,Values=Linux/UNIX Name=network-platform,Values=EC2-VPC Name=instance-type,Values=c4.large

- 자세한 API 내용은 명령 참조DescribeScheduledInstanceAvailability의 섹션을 참조하세요. AWS CLI

### `describe-scheduled-instances`

다음 코드 예시에서는 `describe-scheduled-instances`을 사용하는 방법을 보여 줍니다.

AWS CLI

예약된 인스턴스를 설명하려면

이 예제에서는 지정된 예약 인스턴스를 설명합니다.

명령:

```
aws ec2 describe-scheduled-instances --scheduled-instance-
ids sci-1234-1234-1234-1234-123456789012
```

출력:

```
{
  "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"
      }
  ]
}
```

이 예제에서는 모든 예약된 인스턴스를 설명합니다.

명령:

```
aws ec2 describe-scheduled-instances
```

- 자세한 API 내용은 명령 참조DescribeScheduledInstances의 섹션을 참조하세요. AWS CLI

### describe-security-group-references

다음 코드 예시에서는 describe-security-group-references을 사용하는 방법을 보여 줍니다.

AWS CLI

보안 그룹 참조를 설명하려면

이 예제에서는 에 대한 보안 그룹 참조를 설명합니다sg-bbbb2222. 응답은 보안 그룹이 VPC 의 보안 그룹에서 참조sg-bbbb2222하고 있음을 나타냅니다vpc-aaaaaaaa.

명령:

```
aws ec2 describe-security-group-references --group-id sg-bbbbb22222
```

출력:

```
{
  "SecurityGroupsReferenceSet": [
    {
      "ReferencingVpcId": "vpc-aaaaaaaa ",
      "GroupId": "sg-bbbbb22222",
      "VpcPeeringConnectionId": "pcx-b04deed9"
    }
  ]
}
```

- 자세한 API 내용은 명령 참조DescribeSecurityGroupReferences의 섹션을 참조하세요. AWS CLI

### describe-security-group-rules

다음 코드 예시에서는 describe-security-group-rules을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 보안 그룹에 대한 보안 그룹 규칙을 설명하려면

다음 describe-security-group-rules 예제에서는 지정된 보안 그룹의 보안 그룹 규칙을 설명합니다. filters 옵션을 사용하여 결과를 특정 보안 그룹으로 범위를 지정합니다.

```
aws ec2 describe-security-group-rules \
    --filters Name="group-id",Values="sg-1234567890abcdef0"
```

출력:

```
{
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-abcdef01234567890",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "111122223333",
            "IsEgress": false,
            "IpProtocol": "-1",
            "FromPort": -1,
            "ToPort": -1,
            "ReferencedGroupInfo": {
                "GroupId": "sg-1234567890abcdef0",
                "UserId": "111122223333"
            },
            "Tags": []
        },
        {
            "SecurityGroupRuleId": "sgr-bcdef01234567890a",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "111122223333",
            "IsEgress": true,
            "IpProtocol": "-1",
            "FromPort": -1,
            "ToPort": -1,
            "CidrIpv6": "::/0",
            "Tags": []
        },
        {
            "SecurityGroupRuleId": "sgr-cdef01234567890ab",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "111122223333",
            "IsEgress": true,
            "IpProtocol": "-1",
            "FromPort": -1,
            "ToPort": -1,
```

```
            "CidrIpv4": "0.0.0.0/0",
            "Tags": []
        }
    ]
}
```

예제 2: 보안 그룹 규칙을 설명하려면

다음 `describe-security-group-rules` 예제에서는 지정된 보안 그룹 규칙을 설명합니다.

```
aws ec2 describe-security-group-rules \
    --security-group-rule-ids sgr-cdef01234567890ab
```

출력:

```
{
    "SecurityGroupRules": [
        {
            "SecurityGroupRuleId": "sgr-cdef01234567890ab",
            "GroupId": "sg-1234567890abcdef0",
            "GroupOwnerId": "111122223333",
            "IsEgress": true,
            "IpProtocol": "-1",
            "FromPort": -1,
            "ToPort": -1,
            "CidrIpv4": "0.0.0.0/0",
            "Tags": []
        }
    ]
}
```

자세한 내용은 Amazon VPC 사용 설명서의 보안 그룹 규칙을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeSecurityGroupRules의 섹션을 참조하세요. AWS CLI

### `describe-security-groups`

다음 코드 예시에서는 `describe-security-groups`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 보안 그룹 설명하는 방법

다음 describe-security-groups 예제에서는 지정된 보안 그룹을 설명합니다.

```
aws ec2 describe-security-groups \
    --group-ids sg-903004f8
```

출력:

```
{
    "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",
        }
    ]
}
```

예제 2: 특정 규칙이 있는 보안 그룹을 설명하는 방법

다음 `describe-security-groups` 예제에서는 필터를 사용하여 SSH 트래픽을 허용하는 규칙(포트 22)과 모든 주소의 트래픽을 허용하는 규칙()이 있는 보안 그룹에 결과를 범위를 지정합니다`0.0.0.0/0`. 이 예제에서는 `--query` 파라미터를 사용하여 보안 그룹의 이름만 표시합니다. 보안 그룹이 결과에 반환될 모든 필터와 일치해야 하지만 단일 규칙이 모든 필터와 일치할 필요는 없습니다. 예를 들어 출력은 특정 IP 주소의 SSH 트래픽을 허용하는 규칙과 모든 주소의 HTTP 트래픽을 허용하는 다른 규칙이 있는 보안 그룹을 반환합니다.

```
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
```

출력:

```
default
my-security-group
web-servers
launch-wizard-1
```

예제 3: 태그를 기반으로 보안 그룹을 설명하는 방법

다음 `describe-security-groups` 예제에서는 필터를 사용하여 결과 범위를 보안 그룹 이름에 `test`가 포함되고 `Test=To-delete` 태그가 있는 보안 그룹으로 지정합니다. 이 예제에서는 `--query` 파라미터를 사용하여 IDs 보안 그룹의 이름 및 만 표시합니다.

```
aws ec2 describe-security-groups \
    --filters Name=group-name,Values=*test* Name=tag:Test,Values=To-delete \
    --query "SecurityGroups[*].{Name:GroupName,ID:GroupId}"
```

출력:

```
[
    {
        "Name": "testfornewinstance",
        "ID": "sg-33bb22aa"
    },
    {
        "Name": "newgrouptest",
        "ID": "sg-1a2b3c4d"
    }
]
```

태그 필터를 사용하는 추가 예제는 Amazon EC2 사용 설명서의 태그 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeSecurityGroups의 섹션을 참조하세요. AWS CLI

### `describe-snapshot-attribute`

다음 코드 예시에서는 `describe-snapshot-attribute`을 사용하는 방법을 보여 줍니다.

AWS CLI

스냅샷의 스냅샷 속성을 설명하려면

다음 `describe-snapshot-attribute` 예제에서는 스냅샷이 공유되는 계정을 나열합니다.

```
aws ec2 describe-snapshot-attribute \
    --snapshot-id snap-01234567890abcedf \
    --attribute createVolumePermission
```

출력:

```
{
    "SnapshotId": "snap-01234567890abcedf",
    "CreateVolumePermissions": [
        {
            "UserId": "123456789012"
        }
    ]
}
```

자세한 내용은 Amazon Elastic Compute Cloud 사용 설명서의 Amazon EBS 스냅샷 공유를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeSnapshotAttribute의 섹션을 참조하세요. AWS CLI

### describe-snapshot-tier-status

다음 코드 예시에서는 describe-snapshot-tier-status을 사용하는 방법을 보여 줍니다.

AWS CLI

아카이브된 스냅샷에 대한 아카이브 정보를 보려면

다음 describe-snapshot-tier-status 예제에서는 아카이브된 스냅샷에 대한 아카이브 정보를 제공합니다.

```
aws ec2 describe-snapshot-tier-status \
    --filters "Name=snapshot-id, Values=snap-01234567890abcedf"
```

출력:

```
{
    "SnapshotTierStatuses": [
        {
            "Status": "completed",
            "ArchivalCompleteTime": "2021-09-15T17:33:16.147Z",
            "LastTieringProgress": 100,
            "Tags": [],
            "VolumeId": "vol-01234567890abcedf",
            "LastTieringOperationState": "archival-completed",
            "StorageTier": "archive",
            "OwnerId": "123456789012",
            "SnapshotId": "snap-01234567890abcedf",
```

```
            "LastTieringStartTime": "2021-09-15T16:44:37.574Z"
        }
    ]
}
```

자세한 내용을 알아보려면 Amazon Elastic Compute Cloud 사용 설명서에서 인스턴스 유형을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeSnapshotTierStatus의 섹션을 참조하세요. AWS CLI

### `describe-snapshots`

다음 코드 예시에서는 `describe-snapshots`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 스냅샷을 설명하는 방법

다음 `describe-snapshots` 예제에서는 지정된 스냅샷을 설명합니다.

```
aws ec2 describe-snapshots \
    --snapshot-ids snap-1234567890abcdef0
```

출력:

```
{
    "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"
                }
            ]
```

```
        }
    ]
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon EBS 스냅샷을 참조하세요. EC2

예제 2: 필터를 기반으로 스냅샷을 설명하는 방법

다음 describe-snapshots 예제에서는 필터를 사용하여 pending 상태에 있는 AWS 계정이 소유한 스냅샷으로 결과의 범위를 지정합니다. 이 예제에서는 --query 파라미터를 사용하여 스냅샷 IDs과 스냅샷이 시작된 시간만 표시합니다.

```
aws ec2 describe-snapshots \
    --owner-ids self \
    --filters Name=status,Values=pending \
    --query "Snapshots[*].{ID:SnapshotId,Time:StartTime}"
```

출력:

```
[
    {
        "ID": "snap-1234567890abcdef0",
        "Time": "2019-08-04T12:48:18.000Z"
    },
    {
        "ID": "snap-066877671789bd71b",
        "Time": "2019-08-04T02:45:16.000Z
    },
    ...
]
```

다음 describe-snapshots 예제에서는 필터를 사용하여 결과 범위를 지정된 리전에서 생성된 스냅샷으로 지정합니다. 이 예제에서는 --query 파라미터를 사용하여 스냅샷 만 표시합니다IDs.

```
aws ec2 describe-snapshots \
    --filters Name=volume-id,Values=049df61146c4d7901 \
    --query "Snapshots[*].[SnapshotId]" \
    --output text
```

출력:

```
snap-1234567890abcdef0
```

```
snap-08637175a712c3fb9
...
```

필터를 사용하는 추가 예제는 Amazon EC2 사용 설명서의 리소스 나열 및 필터링을 참조하세요.

예제 3: 태그를 기반으로 스냅샷을 설명하는 방법

다음 `describe-snapshots` 예제에서는 태그 필터를 사용하여 결과 범위를 `Stack=Prod` 태그가 있는 스냅샷으로 지정합니다.

```
aws ec2 describe-snapshots \
    --filters Name=tag:Stack,Values=prod
```

`describe-snapshots` 출력 예제는 예제 1을 참조하세요.

태그 필터를 사용하는 추가 예제는 Amazon EC2 사용 설명서의 태그 작업을 참조하세요.

예제 4: 수명에 기반하여 스냅샷을 설명하는 방법

다음 `describe-snapshots` 예제에서는 JMESPath 표현식을 사용하여 지정된 날짜 이전에 AWS 계정에서 생성된 모든 스냅샷을 설명합니다. 스냅샷만 표시됩니다IDs.

```
aws ec2 describe-snapshots \
    --owner-ids 012345678910 \
    --query "Snapshots[?(StartTime<='2020-03-31')].[SnapshotId]"
```

필터를 사용하는 추가 예제는 Amazon EC2 사용 설명서의 리소스 나열 및 필터링을 참조하세요.

예제 5: 아카이브된 스냅샷만 보는 방법

다음 `describe-snapshots` 예제에서는 아카이브 티어에 저장된 스냅샷만 나열합니다.

```
aws ec2 describe-snapshots \
    --filters "Name=storage-tier,Values=archive"
```

출력:

```
{
    "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": []
        },
    ]
}
```

자세한 내용을 알아보려면 Amazon Elastic Compute Cloud 사용 설명서에서 인스턴스 유형을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeSnapshots의 섹션을 참조하세요. AWS CLI

### `describe-spot-datafeed-subscription`

다음 코드 예시에서는 `describe-spot-datafeed-subscription`을 사용하는 방법을 보여 줍니다.

AWS CLI

계정에 대한 스팟 인스턴스 데이터 피드 구독을 설명하려면

이 예제 명령은 계정의 데이터 피드를 설명합니다.

명령:

```
aws ec2 describe-spot-datafeed-subscription
```

출력:

```
{
    "SpotDatafeedSubscription": {
        "OwnerId": "123456789012",
        "Prefix": "spotdata",
        "Bucket": "my-s3-bucket",
        "State": "Active"
    }
}
```

- 자세한 API 내용은 명령 참조DescribeSpotDatafeedSubscription의 섹션을 참조하세요. AWS CLI

### describe-spot-fleet-instances

다음 코드 예시에서는 describe-spot-fleet-instances을 사용하는 방법을 보여 줍니다.

AWS CLI

스팟 플릿과 연결된 스팟 인스턴스를 설명하려면

이 예제 명령은 지정된 스팟 플릿과 연결된 스팟 인스턴스를 나열합니다.

명령:

```
aws ec2 describe-spot-fleet-instances --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

출력:

```
{
  "ActiveInstances": [
      {
          "InstanceId": "i-1234567890abcdef0",
          "InstanceType": "m3.medium",
          "SpotInstanceRequestId": "sir-08b93456"
      },
      ...
  ],
  "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE"
}
```

- 자세한 API 내용은 명령 참조DescribeSpotFleetInstances의 섹션을 참조하세요. AWS CLI

### describe-spot-fleet-request-history

다음 코드 예시에서는 describe-spot-fleet-request-history을 사용하는 방법을 보여 줍니다.

AWS CLI

스팟 플릿 기록을 설명하려면

이 예제 명령은 지정된 시간에 시작하는 지정된 스팟 플릿에 대한 기록을 반환합니다.

명령:

```
aws ec2 describe-spot-fleet-request-history --spot-fleet-request-id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE --start-time 2015-05-26T00:00:00Z
```

다음 예제 출력은 스팟 플릿에 대한 두 스팟 인스턴스의 성공적인 시작을 보여줍니다.

출력:

```
{
  "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+TfGlWxWWB/Vr5dk5d4LfdgA/DRAHUrYgxzrEXAMPLE=",
  "StartTime": "2015-05-26T00:00:00Z"
}
```

- 자세한 API 내용은 명령 참조DescribeSpotFleetRequestHistory의 섹션을 참조하세요. AWS CLI

### describe-spot-fleet-requests

다음 코드 예시에서는 describe-spot-fleet-requests을 사용하는 방법을 보여 줍니다.

AWS CLI

스팟 플릿 요청을 설명하려면

이 예제에서는 모든 스팟 플릿 요청을 설명합니다.

명령:

```
aws ec2 describe-spot-fleet-requests
```

출력:

```
{
  "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"
    }
  ]
}
```

스팟 플릿 요청을 설명하려면

이 예제에서는 지정된 스팟 플릿 요청을 설명합니다.

명령:

```
aws ec2 describe-spot-fleet-requests --spot-fleet-request-ids sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

출력:

```
{
  "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"
    }
  ]
}
```

- 자세한 API 내용은 명령 참조DescribeSpotFleetRequests의 섹션을 참조하세요. AWS CLI

### `describe-spot-instance-requests`

다음 코드 예시에서는 `describe-spot-instance-requests`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 스팟 인스턴스 요청을 설명하려면

다음 `describe-spot-instance-requests` 예제에서는 지정된 스팟 인스턴스 요청을 설명합니다.

```
aws ec2 describe-spot-instance-requests \
    --spot-instance-request-ids sir-08b93456
```

출력:

```
{
    "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"
        }
    ]
}
```

예제 2: 필터를 기반으로 스팟 인스턴스 요청을 설명하는 방법

다음 describe-spot-instance-requests 예제에서는 필터를 사용하여 지정된 가용 영역에 지정된 인스턴스 유형을 사용하여 결과를 스팟 인스턴스 요청으로 범위를 지정합니다. 이 예제에서는 --query 파라미터를 사용하여 인스턴스 만 표시합니다IDs.

```
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
```

출력:

```
i-057750d42936e468a
i-001efd250faaa6ffa
i-027552a73f021f3bd
...
```

필터를 사용하는 추가 예제는 Amazon Elastic Compute Cloud 사용 설명서의 리소스 나열 및 필터링을 참조하세요.

예제 3: 태그를 기반으로 스팟 인스턴스 요청을 설명하는 방법

다음 describe-spot-instance-requests 예제에서는 태그 필터를 사용하여 태그 가 있는 스팟 인스턴스 요청에 대한 결과의 범위를 지정합니다cost-center=cc123.

```
aws ec2 describe-spot-instance-requests \
    --filters Name=tag:cost-center,Values=cc123
```

describe-spot-instance-requests 출력 예제는 예제 1을 참조하세요.

태그 필터를 사용하는 추가 예제는 Amazon EC2 사용 설명서의 태그 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeSpotInstanceRequests의 섹션을 참조하세요. AWS CLI

### `describe-spot-price-history`

다음 코드 예시에서는 `describe-spot-price-history`을 사용하는 방법을 보여 줍니다.

AWS CLI

스팟 가격 기록을 설명하려면

이 예제 명령은 1월의 특정 날짜에 대한 m1.xlarge 인스턴스의 스팟 가격 기록을 반환합니다.

명령:

```
aws ec2 describe-spot-price-history --instance-types m1.xlarge --start-
time 2014-01-06T07:08:09 --end-time 2014-01-06T08:09:10
```

출력:

```
{
  "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"
```

```
        },
        ...
}
```

Linux/UNIXAmazon의 스팟 가격 기록을 설명하려면 VPC

이 예제 명령은 1월의 특정 날짜에 대한 m1.xlarge, Linux/UNIXAmazon VPC 인스턴스의 스팟 가격 기록을 반환합니다.

명령:

```
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
```

출력:

```
{
  "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"
      }
  ]
}
```

- 자세한 API 내용은 명령 참조DescribeSpotPriceHistory의 섹션을 참조하세요. AWS CLI

### `describe-stale-security-groups`

다음 코드 예시에서는 `describe-stale-security-groups`을 사용하는 방법을 보여 줍니다.

AWS CLI

오래된 보안 그룹을 설명하려면

이 예제에서는 에 대한 오래된 보안 그룹 규칙을 설명합니다vpc-11223344. 응답은 계정의 sg-5fa68d3a에 피어 sg-279ab042에서 참조하는 오래된 수신 SSH 규칙이 VPC있고 계정sg-fe6fba9a의 에 피어 sg-ef6fba8b에서 참조하는 오래된 송신 SSH 규칙이 있음을 보여줍니다 VPC.

명령:

```
aws ec2 describe-stale-security-groups --vpc-id vpc-11223344
```

출력:

```
{
  "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeStaleSecurityGroups의 섹션을 참조하세요. AWS CLI

### describe-store-image-tasks

다음 코드 예시에서는 describe-store-image-tasks을 사용하는 방법을 보여 줍니다.

AWS CLI

AMI 스토어 작업의 진행 상황을 설명하려면

다음 describe-store-image-tasks 예제에서는 AMI 스토어 작업의 진행 상황을 설명합니다.

```
aws ec2 describe-store-image-tasks
```

출력:

```
{
    "StoreImageTaskResults": [
        {
            "AmiId": "ami-1234567890abcdef0",
            "Bucket": "my-ami-bucket",
```

```
            "ProgressPercentage": 17,
            "S3objectKey": "ami-1234567890abcdef0.bin",
            "StoreTaskState": "InProgress",
            "StoreTaskFailureReason": null,
            "TaskStartTime": "2022-01-01T01:01:01.001Z"
        }
    ]
}
```

S3를 AMI 사용하여 를 저장하고 복원하는 방법에 대한 자세한 내용은 Amazon EC2 사용 설명서의 S3 <https://docs.aws.amazon.com/AWS EC2/latest/UserGuide/ami-store-restore.html>을 AMI 사용하여 를 저장하고 복원을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeStoreImageTasks의 섹션을 참조하세요. AWS CLI

### `describe-subnets`

다음 코드 예시에서는 `describe-subnets`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 모든 서브넷을 설명하는 방법

다음 `describe-subnets` 예제에서는 서브넷의 세부 정보를 표시합니다.

```
aws ec2 describe-subnets
```

출력:

```
{
    "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
            }
        }
    ]
```

```
}
```

자세한 내용은 AWS VPC 사용 설명서의 VPCs 및 서브넷 작업을 참조하세요.

예제 2: 특정 의 서브넷 설명 VPC

다음 `describe-subnets` 예제에서는 필터를 사용하여 지정된 의 서브넷에 대한 세부 정보를 검색합니다VPC.

```
aws ec2 describe-subnets \
    --filters "Name=vpc-id,Values=vpc-3EXAMPLE"
```

출력:

```
{
    "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
            }
        }
    ]
}
```

자세한 내용은 AWS VPC 사용 설명서의 VPCs 및 서브넷 작업을 참조하세요.

예제 3: 특정 태그의 서브넷을 설명하는 방법

다음 describe-subnets 예제에서는 필터를 사용하여 태그CostCenter=123가 있는 서브넷의 세부 정보와 --query 파라미터를 검색하여 이 태그가 있는 서브넷IDs의 서브넷을 표시합니다.

```
aws ec2 describe-subnets \
    --filters "Name=tag:CostCenter,Values=123" \
    --query "Subnets[*].SubnetId" \
    --output text
```

출력:

```
subnet-0987a87c8b37348ef
subnet-02a95061c45f372ee
subnet-03f720e7de2788d73
```

자세한 내용은 Amazon VPC 사용 설명서의 VPCs 및 서브넷 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeSubnets의 섹션을 참조하세요. AWS CLI

### describe-tags

다음 코드 예시에서는 describe-tags을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 단일 리소스에 대한 모든 태그를 설명하려면

다음 describe-tags 예제에서는 지정된 인스턴스의 태그를 설명합니다.

```
aws ec2 describe-tags \
    --filters "Name=resource-id,Values=i-1234567890abcdef8"
```

출력:

```
{
    "Tags": [
        {
            "ResourceType": "instance",
            "ResourceId": "i-1234567890abcdef8",
            "Value": "Test",
            "Key": "Stack"
        },
        {
            "ResourceType": "instance",
            "ResourceId": "i-1234567890abcdef8",
            "Value": "Beta Server",
            "Key": "Name"
        }
    ]
}
```

예제 2: 리소스 유형에 대한 모든 태그를 설명하려면

다음 `describe-tags` 예제에서는 볼륨에 대한 태그를 설명합니다.

```
aws ec2 describe-tags \
    --filters "Name=resource-type,Values=volume"
```

출력:

```
{
    "Tags": [
        {
            "ResourceType": "volume",
            "ResourceId": "vol-1234567890abcdef0",
            "Value": "Project1",
            "Key": "Purpose"
        },
        {
            "ResourceType": "volume",
            "ResourceId": "vol-049df61146c4d7901",
            "Value": "Logs",
            "Key": "Purpose"
        }
    ]
}
```

예제 3: 모든 태그를 설명하려면

다음 describe-tags 예제에서는 모든 리소스의 태그를 설명합니다.

```
aws ec2 describe-tags
```

예제 4: 태그 키를 기반으로 리소스에 대한 태그를 설명하는 방법

다음 describe-tags 예제에서는 키 가 있는 태그가 있는 리소스의 태그를 설명합니다Stack.

```
aws ec2 describe-tags \
    --filters Name=key,Values=Stack
```

출력:

```
{
    "Tags": [
        {
            "ResourceType": "volume",
            "ResourceId": "vol-027552a73f021f3b",
            "Value": "Production",
            "Key": "Stack"
        },
        {
            "ResourceType": "instance",
            "ResourceId": "i-1234567890abcdef8",
            "Value": "Test",
            "Key": "Stack"
        }
    ]
}
```

예제 5: 태그 키 및 태그 값을 기반으로 리소스의 태그를 설명하는 방법

다음 describe-tags 예제에서는 태그가 있는 리소스의 태그를 설명합니다Stack=Test.

```
aws ec2 describe-tags \
    --filters Name=key,Values=Stack Name=value,Values=Test
```

출력:

```
{
```

```
    "Tags": [
        {
            "ResourceType": "image",
            "ResourceId": "ami-3ac336533f021f3bd",
            "Value": "Test",
            "Key": "Stack"
        },
        {
            "ResourceType": "instance",
            "ResourceId": "i-1234567890abcdef8",
            "Value": "Test",
            "Key": "Stack"
        }
    ]
}
```

다음 describe-tags 예제에서는 대체 구문을 사용하여 태그가 인 리소스를 설명합니다Stack=Test.

```
aws ec2 describe-tags \
    --filters "Name=tag:Stack,Values=Test"
```

다음 describe-tags 예제에서는 키가 Purpose 있고 값이 없는 태그가 있는 모든 인스턴스의 태그를 설명합니다.

```
aws ec2 describe-tags \
    --filters "Name=resource-
type,Values=instance" "Name=key,Values=Purpose" "Name=value,Values="
```

출력:

```
{
    "Tags": [
        {
            "ResourceType": "instance",
            "ResourceId": "i-1234567890abcdef5",
            "Value": null,
            "Key": "Purpose"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeTags의 섹션을 참조하세요. AWS CLI

### `describe-traffic-mirror-filters`

다음 코드 예시에서는 `describe-traffic-mirror-filters`을 사용하는 방법을 보여 줍니다.

AWS CLI

트래픽 미러 필터를 보려면

다음 `describe-traffic-mirror-filters` 예제에서는 모든 트래픽 미러 필터에 대한 세부 정보를 표시합니다.

```
aws ec2 describe-traffic-mirror-filters
```

출력:

```
{
    "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": []
        }
    ]
}
```

자세한 내용은 트래픽 미러링 가이드의 트래픽 미러 필터 보기를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeTrafficMirrorFilters의 섹션을 참조하세요. AWS CLI

### describe-traffic-mirror-sessions

다음 코드 예시에서는 describe-traffic-mirror-sessions을 사용하는 방법을 보여 줍니다.

AWS CLI

트래픽 미러 세션을 설명하려면

다음 describe-traffic-mirror-sessions 예제에서는 Traffic Mirror 세션에 대한 세부 정보를 표시합니다.

```
aws ec2 describe-traffic-mirror-sessions
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 트래픽 미러링 가이드의 트래픽 미러 세션 세부 정보 보기를 참조하세요. AWS

- 자세한 API 내용은 명령 참조DescribeTrafficMirrorSessions의 섹션을 참조하세요. AWS CLI

### describe-traffic-mirror-targets

다음 코드 예시에서는 describe-traffic-mirror-targets을 사용하는 방법을 보여 줍니다.

AWS CLI

트래픽 미러 대상을 설명하려면

다음 describe-traffic-mirror-targets 예제에서는 지정된 트래픽 미러 대상에 대한 정보를 표시합니다.

```
aws ec2 describe-traffic-mirror-targets \
    --traffic-mirror-target-ids tmt-0dabe9b0a6EXAMPLE
```

출력:

```
{
    "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": []
        }
    ]
}
```

자세한 내용은 Amazon 트래픽 미러링 가이드의 트래픽 미러 대상을 참조하세요. VPC

- 자세한 API 내용은 명령 참조DescribeTrafficMirrorTargets의 섹션을 참조하세요. AWS CLI

### describe-transit-gateway-attachments

다음 코드 예시에서는 describe-transit-gateway-attachments을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 첨부 파일을 보려면

다음 describe-transit-gateway-attachments 예제에서는 전송 게이트웨이 연결에 대한 세부 정보를 표시합니다.

```
aws ec2 describe-transit-gateway-attachments
```

출력:

```
{
    "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": []
        }
```

```
    ]
}
```

자세한 내용은 Transit Gateways 가이드의 전송 게이트웨이 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeTransitGatewayAttachments의 섹션을 참조하세요. AWS CLI

### describe-transit-gateway-connect-peers

다음 코드 예시에서는 describe-transit-gateway-connect-peers을 사용하는 방법을 보여 줍니다.

AWS CLI

Transit Gateway Connect 피어를 설명하려면

다음 describe-transit-gateway-connect-peers 예제에서는 지정된 Connect 피어를 설명합니다.

```
aws ec2 describe-transit-gateway-connect-peers \
    --transit-gateway-connect-peer-ids tgw-connect-peer-0666adbac4EXAMPLE
```

출력:

```
{
    "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": []
    }
]
}
```

자세한 내용은 Transit Gateways 가이드의 Transit Gateway Connect 연결 및 Transit Gateway Connect 피어를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeTransitGatewayConnectPeers의 섹션을 참조하세요. AWS CLI

### `describe-transit-gateway-connects`

다음 코드 예시에서는 `describe-transit-gateway-connects`을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 Connect 연결 설명

다음 `describe-transit-gateway-connects` 예제에서는 지정된 Connect 연결에 대해 설명합니다.

```
aws ec2 describe-transit-gateway-connects \
    --transit-gateway-attachment-ids tgw-attach-037012e5dcEXAMPLE
```

출력:

```
{
```

```
    "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": []
        }
    ]
}
```

자세한 내용은 Transit Gateways 가이드의 Transit Gateway Connect 연결 및 Transit Gateway Connect 피어를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeTransitGatewayConnects의 섹션을 참조하세요. AWS CLI

### describe-transit-gateway-multicast-domains

다음 코드 예시에서는 describe-transit-gateway-multicast-domains을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 멀티캐스트 도메인을 설명하려면

다음 describe-transit-gateway-multicast-domains 예제에서는 모든 전송 게이트웨이 멀티캐스트 도메인에 대한 세부 정보를 표시합니다.

```
aws ec2 describe-transit-gateway-multicast-domains
```

출력:

```
{

    "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": {
                "Igmpv2Support": "disable",
                "StaticSourcesSupport": "enable",
                "AutoAcceptSharedAssociations": "disable"
            },
            "State": "available",
            "CreationTime": "2019-12-10T18:32:50+00:00",
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "mc1"
                }
            ]
        }
    ]
}
```

자세한 내용은 Transit Gateways 가이드의 멀티캐스트 도메인 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeTransitGatewayMulticastDomains의 섹션을 참조하세요. AWS CLI

### describe-transit-gateway-peering-attachments

다음 코드 예시에서는 describe-transit-gateway-peering-attachments을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 피어링 연결 설명

다음 describe-transit-gateway-peering-attachments 예제에서는 모든 전송 게이트웨이 피어링 연결에 대한 세부 정보를 표시합니다.

```
aws ec2 describe-transit-gateway-peering-attachments
```

출력:

```
{
```

```
    "TransitGatewayPeeringAttachments": [
        {
            "TransitGatewayAttachmentId": "tgw-attach-4455667788aabbccd",
            "RequesterTgwInfo": {
                "TransitGatewayId": "tgw-123abc05e04123abc",
                "OwnerId": "123456789012",
                "Region": "us-west-2"
            },
            "AccepterTgwInfo": {
                "TransitGatewayId": "tgw-11223344aabbcc112",
                "OwnerId": "123456789012",
                "Region": "us-east-2"
            },
            "State": "pendingAcceptance",
            "CreationTime": "2019-12-09T11:38:05.000Z",
            "Tags": []
        }
    ]
}
```

자세한 내용은 Transit Gateways 가이드의 Transit Gateway 피어링 연결을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeTransitGatewayPeeringAttachments의 섹션을 참조하세요. AWS CLI

### describe-transit-gateway-policy-tables

다음 코드 예시에서는 describe-transit-gateway-policy-tables을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 정책 테이블을 설명하려면

다음 describe-transit-gateway-policy-tables 예제에서는 지정된 전송 게이트웨이 정책 테이블에 대해 설명합니다.

```
aws ec2 describe-transit-gateway-policy-tables \
    --transit-gateway-policy-table-ids tgw-ptb-0a16f134b78668a81
```

출력:

```
{
```

```
    "TransitGatewayPolicyTables": [
        {
            "TransitGatewayPolicyTableId": "tgw-ptb-0a16f134b78668a81",
            "TransitGatewayId": "tgw-067f8505c18f0bd6e",
            "State": "available",
            "CreationTime": "2023-11-28T16:36:43+00:00",
            "Tags": []
        }
    ]
}
```

자세한 내용은 Transit Gateway 사용 설명서의 Transit Gateway 정책 테이블을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeTransitGatewayPolicyTables의 섹션을 참조하세요. AWS CLI

### describe-transit-gateway-route-tables

다음 코드 예시에서는 describe-transit-gateway-route-tables을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 라우팅 테이블을 설명하려면

다음 describe-transit-gateway-route-tables 예제에서는 전송 게이트웨이 라우팅 테이블에 대한 세부 정보를 표시합니다.

```
aws ec2 describe-transit-gateway-route-tables
```

출력:

```
{
    "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": []
        }
    ]
}
```

자세한 내용은 Transit Gateways 가이드의 전송 게이트웨이 라우팅 테이블 보기를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeTransitGatewayRouteTables의 섹션을 참조하세요. AWS CLI

### describe-transit-gateway-vpc-attachments

다음 코드 예시에서는 describe-transit-gateway-vpc-attachments을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 VPC 연결 설명

다음 describe-transit-gateway-vpc-attachments 예제에서는 전송 게이트웨이 VPC 연결에 대한 세부 정보를 표시합니다.

```
aws ec2 describe-transit-gateway-vpc-attachments
```

출력:

```
{
    "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"
                }
            ]
        }
    ]
}
```

자세한 내용은 Transit Gateways 가이드의 VPC 첨부 파일 보기를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeTransitGatewayVpcAttachments의 섹션을 참조하세요. AWS CLI

### describe-transit-gateways

다음 코드 예시에서는 describe-transit-gateways을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이를 설명하려면

다음 describe-transit-gateways 예제에서는 전송 게이트웨이에 대한 세부 정보를 검색합니다.

```
aws ec2 describe-transit-gateways
```

출력:

```
{
    "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"
                }
            ]
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeTransitGateways의 섹션을 참조하세요. AWS CLI

### describe-verified-access-endpoints

다음 코드 예시에서는 describe-verified-access-endpoints을 사용하는 방법을 보여 줍니다.

AWS CLI

Verified Access 엔드포인트를 설명하려면

다음 delete-verified-access-endpoints 예제에서는 지정된 Verified Access 엔드포인트를 설명합니다.

```
aws ec2 describe-verified-access-endpoints \
    --verified-access-endpoint-ids vae-066fac616d4d546f2
```

출력:

```
{
    "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"
                }
            ]
        }
    ]
}
```

자세한 내용은 Verified Access 사용 설명서의 Verified Access 엔드포인트를 참조하세요. AWS

- 자세한 API 내용은 명령 참조DescribeVerifiedAccessEndpoints의 섹션을 참조하세요. AWS CLI

### `describe-verified-access-groups`

다음 코드 예시에서는 `describe-verified-access-groups`을 사용하는 방법을 보여 줍니다.

AWS CLI

Verified Access 그룹을 설명하려면

다음 `describe-verified-access-groups` 예제에서는 지정된 Verified Access 그룹에 대해 설명합니다.

```
aws ec2 describe-verified-access-groups \
    --verified-access-group-ids vagr-0dbe967baf14b7235
```

출력:

```
{
    "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"
                }
            ]
        }
    ]
}
```

자세한 내용은 Verified Access 사용 설명서의 Verified Access 그룹을 참조하세요. AWS

- 자세한 API 내용은 명령 참조DescribeVerifiedAccessGroups의 섹션을 참조하세요. AWS CLI

### `describe-verified-access-instance-logging-configurations`

다음 코드 예시에서는 `describe-verified-access-instance-logging-configurations`을 사용하는 방법을 보여 줍니다.

AWS CLI

Verified Access 인스턴스의 로깅 구성을 설명하려면

다음 `describe-verified-access-instance-logging-configurations` 예제에서는 지정된 Verified Access 인스턴스의 로깅 구성을 설명합니다.

```
aws ec2 describe-verified-access-instance-logging-configurations \
    --verified-access-instance-ids vai-0ce000c0b7643abea
```

출력:

```
{
    "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
            }
        }
    ]
}
```

자세한 내용은 Verified Access 사용 설명서의 Verified Access 로그를 참조하세요. AWS

- 자세한 API 내용은 명령 참조DescribeVerifiedAccessInstanceLoggingConfigurations의 섹션을 참조하세요. AWS CLI

### describe-verified-access-instances

다음 코드 예시에서는 describe-verified-access-instances을 사용하는 방법을 보여 줍니다.

AWS CLI

Verified Access 인스턴스를 설명하려면

다음 describe-verified-access-instances 예제에서는 지정된 Verified Access 인스턴스를 설명합니다.

```
aws ec2 describe-verified-access-instances \
    --verified-access-instance-ids vai-0ce000c0b7643abea
```

출력:

```
{
    "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"
                }
            ]
        }
    ]
}
```

자세한 내용은 Verified Access 사용 설명서의 Verified Access 인스턴스를 참조하세요. AWS

- 자세한 API 내용은 명령 참조DescribeVerifiedAccessInstances의 섹션을 참조하세요. AWS CLI

### describe-verified-access-trust-providers

다음 코드 예시에서는 describe-verified-access-trust-providers을 사용하는 방법을 보여 줍니다.

AWS CLI

Verified Access 신뢰 공급자를 설명하려면

다음 describe-verified-access-trust-providers 예제에서는 지정된 Verified Access 신뢰 공급자를 설명합니다.

```
aws ec2 describe-verified-access-trust-providers \
    --verified-access-trust-provider-ids vatp-0bb32de759a3e19e7
```

출력:

```
{
    "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"
                }
            ]
        }
    ]
}
```

자세한 내용은 Verified Access 사용 설명서의 Verified Access에 대한 신뢰 공급자를 참조하세요. AWS

- 자세한 API 내용은 명령 참조DescribeVerifiedAccessTrustProviders의 섹션을 참조하세요. AWS CLI

### `describe-volume-attribute`

다음 코드 예시에서는 `describe-volume-attribute`을 사용하는 방법을 보여 줍니다.

AWS CLI

볼륨 속성을 설명하려면

이 예제 명령은 ID 를 사용하여 볼륨의 `autoEnableIo` 속성을 설명합니다`vol-049df61146c4d7901`.

명령:

```
aws ec2 describe-volume-attribute --volume-id vol-049df61146c4d7901 --
attribute autoEnableIO
```

출력:

```
{
    "AutoEnableIO": {
        "Value": false
    },
    "VolumeId": "vol-049df61146c4d7901"
}
```

- 자세한 API 내용은 명령 참조DescribeVolumeAttribute의 섹션을 참조하세요. AWS CLI

### `describe-volume-status`

다음 코드 예시에서는 `describe-volume-status`을 사용하는 방법을 보여 줍니다.

AWS CLI

단일 볼륨의 상태를 설명하려면

이 예제 명령은 볼륨 의 상태를 설명합니다`vol-1234567890abcdef0`.

명령:

```
aws ec2 describe-volume-status --volume-ids vol-1234567890abcdef0
```

출력:

```
{
    "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": []
        }
    ]
}
```

손상된 볼륨의 상태를 설명하려면

이 예제 명령은 손상된 모든 볼륨의 상태를 설명합니다. 이 예제 출력에서는 손상된 볼륨이 없습니다.

명령:

```
aws ec2 describe-volume-status --filters Name=volume-status.status,Values=impaired
```

출력:

```
{
    "VolumeStatuses": []
}
```

상태 확인에 실패한 볼륨이 있는 경우(상태가 손상됨) Amazon EC2 사용 설명서의 손상된 볼륨 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeVolumeStatus의 섹션을 참조하세요. AWS CLI

### describe-volumes-modifications

다음 코드 예시에서는 describe-volumes-modifications을 사용하는 방법을 보여 줍니다.

AWS CLI

볼륨의 수정 상태를 설명하려면

다음 describe-volumes-modifications 예제에서는 지정된 볼륨의 볼륨 수정 상태를 설명합니다.

```
aws ec2 describe-volumes-modifications \
    --volume-ids vol-1234567890abcdef0
```

출력:

```
{
    "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
    }
}
```

- 자세한 API 내용은 명령 참조DescribeVolumesModifications의 섹션을 참조하세요. AWS CLI

### `describe-volumes`

다음 코드 예시에서는 `describe-volumes`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 볼륨 설명

다음 `describe-volumes` 예제에서는 현재 리전에서 지정된 볼륨을 설명합니다.

```
aws ec2 describe-volumes \
    --volume-ids vol-049df61146c4d7901 vol-1234567890abcdef0
```

출력:

```
{
    "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
        }
    ]
}
```

예제 2: 특정 인스턴스에 연결된 볼륨 설명

다음 `describe-volumes` 예제에서는 지정된 인스턴스에 연결되고 인스턴스가 종료될 때 삭제하도록 설정된 모든 볼륨을 설명합니다.

```
aws ec2 describe-volumes \
    --region us-east-1 \
    --filters Name=attachment.instance-
id,Values=i-1234567890abcdef0 Name=attachment.delete-on-termination,Values=true
```

`describe-volumes` 출력 예제는 예제 1을 참조하세요.

예제 3: 특정 가용 영역에서 사용 가능한 볼륨 설명

다음 `describe-volumes` 예제에서는 상태가 `available` 이고 지정된 가용 영역에 있는 모든 볼륨을 설명합니다.

```
aws ec2 describe-volumes \
```

```
    --filters Name=status,Values=available Name=availability-zone,Values=us-east-1a
```

describe-volumes 출력 예제는 예제 1을 참조하세요.

예제 4: 태그를 기반으로 볼륨 설명

다음 describe-volumes 예제에서는 태그 키가 있는 모든 볼륨Name과 로 시작하는 값을 설명합니다Test. 그런 다음 출력은 태그와 볼륨만 표시하는 쿼리IDs로 필터링됩니다.

```
aws ec2 describe-volumes \
    --filters Name=tag:Name,Values=Test* \
    --query "Volumes[*].{ID:VolumeId,Tag:Tags}"
```

출력:

```
[
    {
        "Tag": [
            {
                "Value": "Test2",
                "Key": "Name"
            }
        ],
        "ID": "vol-1234567890abcdef0"
    },
    {
        "Tag": [
            {
                "Value": "Test1",
                "Key": "Name"
            }
        ],
        "ID": "vol-049df61146c4d7901"
    }
]
```

태그 필터를 사용하는 추가 예제는 Amazon EC2 사용 설명서의 태그 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeVolumes의 섹션을 참조하세요. AWS CLI

### describe-vpc-attribute

다음 코드 예시에서는 describe-vpc-attribute을 사용하는 방법을 보여 줍니다.

AWS CLI

enableDnsSupport 속성을 설명하려면

이 예제에서는 enableDnsSupport 속성을 설명합니다. 이 속성은 에 대한 DNS 해상도가 활성화되어 있는지 여부를 나타냅니다VPC. 이 속성이 인 경우 Amazon DNS 서버는 인스턴스true의 DNS 호스트 이름을 해당 IP 주소로 확인합니다. 그렇지 않으면 확인하지 않습니다.

명령:

```
aws ec2 describe-vpc-attribute --vpc-id vpc-a01106c2 --attribute enableDnsSupport
```

출력:

```
{
    "VpcId": "vpc-a01106c2",
    "EnableDnsSupport": {
        "Value": true
    }
}
```

enableDnsHostnames 속성을 설명하려면

이 예제에서는 enableDnsHostnames 속성을 설명합니다. 이 속성은 인스턴스가 DNS 호스트 이름 VPC 가져오기에서 시작되었는지 여부를 나타냅니다. 이 속성이 true인 경우 DNS 호스트 이름 VPC 가져오기의 인스턴스입니다. 그렇지 않으면 인스턴스가 되지 않습니다.

명령:

```
aws ec2 describe-vpc-attribute --vpc-id vpc-a01106c2 --attribute enableDnsHostnames
```

출력:

```
{
    "VpcId": "vpc-a01106c2",
    "EnableDnsHostnames": {
        "Value": true
    }
}
```

- 자세한 API 내용은 명령 참조DescribeVpcAttribute의 섹션을 참조하세요. AWS CLI

### describe-vpc-classic-link-dns-support

다음 코드 예시에서는 describe-vpc-classic-link-dns-support을 사용하는 방법을 보여 줍니다.

AWS CLI

에 대한 지원을 설명 ClassicLink DNS하려면 VPCs

이 예제에서는 모든 의 지원 상태를 설명합니다 ClassicLink DNSVPCs.

명령:

```
aws ec2 describe-vpc-classic-link-dns-support
```

출력:

```
{
  "Vpcs": [
    {
      "VpcId": "vpc-88888888",
      "ClassicLinkDnsSupported": true
    },
    {
      "VpcId": "vpc-1a2b3c4d",
      "ClassicLinkDnsSupported": false
    }
  ]
}
```

- 자세한 API 내용은 명령 참조DescribeVpcClassicLinkDnsSupport의 섹션을 참조하세요. AWS CLI

### describe-vpc-classic-link

다음 코드 예시에서는 describe-vpc-classic-link을 사용하는 방법을 보여 줍니다.

AWS CLI

의 ClassicLink 상태를 설명하려면 VPCs

이 예제에서는 vpc-88888888의 ClassicLink 상태를 나열합니다.

명령:

```
aws ec2 describe-vpc-classic-link --vpc-id vpc-88888888
```

출력:

```
{
  "Vpcs": [
    {
      "ClassicLinkEnabled": true,
      "VpcId": "vpc-88888888",
      "Tags": [
        {
          "Value": "classiclinkvpc",
          "Key": "Name"
        }
      ]
    }
  ]
}
```

이 예제VPCs에서는 Classiclink에 대해 활성화된 만 나열`is-classic-link-enabled`합니다(의 필터 값은 로 설정됨`true`).

명령:

```
aws ec2 describe-vpc-classic-link --filter "Name=is-classic-link-
enabled,Values=true"
```

- 자세한 API 내용은 명령 참조DescribeVpcClassicLink의 섹션을 참조하세요. AWS CLI

### `describe-vpc-endpoint-connection-notifications`

다음 코드 예시에서는 `describe-vpc-endpoint-connection-notifications`을 사용하는 방법을 보여 줍니다.

AWS CLI

엔드포인트 연결 알림을 설명하려면

다음 `describe-vpc-endpoint-connection-notifications` 예제에서는 모든 엔드포인트 연결 알림을 설명합니다.

```
aws ec2 describe-vpc-endpoint-connection-notifications
```

출력:

```
{
   "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"
       }
   ]
 }
```

- 자세한 API 내용은 명령 참조DescribeVpcEndpointConnectionNotifications의 섹션을 참조하세요. AWS CLI

### describe-vpc-endpoint-connections

다음 코드 예시에서는 describe-vpc-endpoint-connections을 사용하는 방법을 보여 줍니다.

AWS CLI

VPC 엔드포인트 연결을 설명하려면

이 예제에서는 엔드포인트 서비스에 대한 인터페이스 엔드포인트 연결을 설명하고 결과를 필터링하여 인 엔드포인트를 표시합니다PendingAcceptance.

명령:

```
aws ec2 describe-vpc-endpoint-connections --filters Name=vpc-endpoint-
state,Values=pendingAcceptance
```

출력:

```
{
  "VpcEndpointConnections": [
      {
          "VpcEndpointId": "vpce-0abed31004e618123",
          "ServiceId": "vpce-svc-0abced088d20def56",
          "CreationTimestamp": "2017-11-30T10:00:24.350Z",
          "VpcEndpointState": "pendingAcceptance",
          "VpcEndpointOwner": "123456789012"
      }
  ]
}
```

- 자세한 API 내용은 명령 참조DescribeVpcEndpointConnections의 섹션을 참조하세요. AWS CLI

### `describe-vpc-endpoint-service-configurations`

다음 코드 예시에서는 `describe-vpc-endpoint-service-configurations`을 사용하는 방법을 보여 줍니다.

AWS CLI

엔드포인트 서비스 구성을 설명하려면

다음 `describe-vpc-endpoint-service-configurations` 예제에서는 엔드포인트 서비스 구성을 설명합니다.

```
aws ec2 describe-vpc-endpoint-service-configurations
```

출력:

```
{
    "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/GWLBService/123210844e429123"
            ],
            "Tags": []
        },
        {
            "ServiceType": [
                {
                    "ServiceType": "Interface"
                }
            ],
            "ServiceId": "vpce-svc-123cabc125efa123",
            "ServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-123cabc125efa123",
            "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-123cabc125efa123.us-east-1.vpce.amazonaws.com"
            ],
            "PrivateDnsName": "example.com",
            "PrivateDnsNameConfiguration": {
                "State": "failed",
                "Type": "TXT",
                "Value": "vpce:qUAth3FdeABCApUiXabc",
                "Name": "_1d367jvbg34znqvyefrj"
            },
            "Tags": []
        }
```

```
    ]
}
```

자세한 내용은 Amazon VPC 사용 설명서의 VPC 엔드포인트 서비스를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeVpcEndpointServiceConfigurations의 섹션을 참조하세요. AWS CLI

### `describe-vpc-endpoint-service-permissions`

다음 코드 예시에서는 `describe-vpc-endpoint-service-permissions`을 사용하는 방법을 보여 줍니다.

AWS CLI

엔드포인트 서비스 권한을 설명하려면

이 예제에서는 지정된 엔드포인트 서비스에 대한 권한을 설명합니다.

명령:

```
aws ec2 describe-vpc-endpoint-service-permissions --service-id vpce-svc-03d5ebb7d9579a2b3
```

출력:

```
{
   "AllowedPrincipals": [
       {
           "PrincipalType": "Account",
           "Principal": "arn:aws:iam::123456789012:root"
       }
   ]
}
```

- 자세한 API 내용은 명령 참조DescribeVpcEndpointServicePermissions의 섹션을 참조하세요. AWS CLI

### `describe-vpc-endpoint-services`

다음 코드 예시에서는 `describe-vpc-endpoint-services`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 모든 VPC 엔드포인트 서비스를 설명하려면

다음 “describe-vpc-endpoint-services” 예제에서는 AWS 리전에 대한 모든 VPC 엔드포인트 서비스를 나열합니다.

```
aws ec2 describe-vpc-endpoint-services
```

출력:

```
{
    "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"
    ]
}
```

자세한 내용은 사용 설명서의 사용 가능한 AWS 서비스 이름 보기를 참조하세요. AWS PrivateLink

예제 2: 엔드포인트 서비스에 대한 세부 정보를 설명하려면

다음 “describe-vpc-endpoint-services” 예제에서는 Amazon S3 인터페이스 엔드포인트 서비스의 세부 정보를 나열합니다.

```
aws ec2 describe-vpc-endpoint-services \
    --filter "Name=service-type,Values=Interface" Name=service-
name,Values=com.amazonaws.us-east-1.s3
```

출력:

```
{
    "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"
    ]
}
```

자세한 내용은 사용 설명서의 사용 가능한 AWS 서비스 이름 보기를 참조하세요. AWS PrivateLink

- 자세한 API 내용은 명령 참조DescribeVpcEndpointServices의 섹션을 참조하세요. AWS CLI

### `describe-vpc-endpoints`

다음 코드 예시에서는 `describe-vpc-endpoints`을 사용하는 방법을 보여 줍니다.

AWS CLI

VPC 엔드포인트를 설명하려면

다음 `describe-vpc-endpoints` 예제에서는 모든 VPC 엔드포인트에 대한 세부 정보를 표시합니다.

```
aws ec2 describe-vpc-endpoints
```

출력:

```
{
    "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      \"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"
        }
    ]
}
```

자세한 내용은 Amazon VPC 사용 설명서의 VPC 엔드포인트를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeVpcEndpoints의 섹션을 참조하세요. AWS CLI

### `describe-vpc-peering-connections`

다음 코드 예시에서는 `describe-vpc-peering-connections`을 사용하는 방법을 보여 줍니다.

AWS CLI

VPC 피어링 연결을 설명하려면

이 예제에서는 모든 VPC 피어링 연결을 설명합니다.

명령:

```
aws ec2 describe-vpc-peering-connections
```

출력:

```
{
    "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"
            }
        }
    ]
}
```

특정 VPC 피어링 연결을 설명하려면

이 예제에서는 보류 중인 허용 상태에 있는 모든 VPC 피어링 연결을 설명합니다.

명령:

```
aws ec2 describe-vpc-peering-connections --filters Name=status-code,Values=pending-acceptance
```

이 예제에서는 Owner=Finance 태그가 있는 모든 VPC 피어링 연결을 설명합니다.

명령:

```
aws ec2 describe-vpc-peering-connections --filters Name=tag:Owner,Values=Finance
```

이 예제에서는 지정된 VPC, vpc-1a2b3c4d에 대해 요청한 모든 VPC 피어링 연결을 설명합니다.

명령:

```
aws ec2 describe-vpc-peering-connections --filters Name=requester-vpc-info.vpc-id,Values=vpc-1a2b3c4d
```

- 자세한 API 내용은 명령 참조DescribeVpcPeeringConnections의 섹션을 참조하세요. AWS CLI

### `describe-vpcs`

다음 코드 예시에서는 `describe-vpcs`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 모든 를 설명하려면 VPCs

다음 `describe-vpcs` 예제에서는 에 대한 세부 정보를 검색합니다VPCs.

```
aws ec2 describe-vpcs
```

출력:

```
{
    "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"
                }
            ]
        }
    ]
}
```

예제 2: 지정된 VPC

다음 `describe-vpcs` 예제에서는 지정된 에 대한 세부 정보를 검색합니다VPC.

```
aws ec2 describe-vpcs \
    --vpc-ids vpc-06e4ab6c6cEXAMPLE
```

출력:

```
{
    "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"
                }
            ]
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeVpcs의 섹션을 참조하세요. AWS CLI

### `describe-vpn-connections`

다음 코드 예시에서는 `describe-vpn-connections`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: VPN 연결 설명

다음 `describe-vpn-connections` 예제에서는 모든 Site-to-Site VPN 연결을 설명합니다.

```
aws ec2 describe-vpn-connections
```

출력:

```
{
    "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": ""
                }
            ]
        }
    ]
}
```

자세한 내용은 AWS Site-to-Site VPN 사용 설명서의 작동 방식을 AWS Site-to-Site VPN 참조하세요.

예제 2: 사용 가능한 VPN 연결을 설명하려면

다음 `describe-vpn-connections` 예제에서는 상태와의 VPN 연결을 설명합니다 Site-to-Site`available`.

```
aws ec2 describe-vpn-connections \
    --filters "Name=state,Values=available"
```

자세한 내용은 AWS Site-to-Site VPN 사용 설명서의 작동 방식을 AWS Site-to-Site VPN 참조하세요.

- 자세한 API 내용은 명령 참조DescribeVpnConnections의 섹션을 참조하세요. AWS CLI

### `describe-vpn-gateways`

다음 코드 예시에서는 `describe-vpn-gateways`을 사용하는 방법을 보여 줍니다.

AWS CLI

가상 프라이빗 게이트웨이를 설명하려면

이 예제에서는 가상 프라이빗 게이트웨이에 대해 설명합니다.

명령:

```
aws ec2 describe-vpn-gateways
```

출력:

```
{
    "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"
                }
```

```
            ]
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeVpnGateways의 섹션을 참조하세요. AWS CLI

### `detach-classic-link-vpc`

다음 코드 예시에서는 `detach-classic-link-vpc`을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 EC2-Classic 인스턴스의 연결을 해제(분리)하려면 VPC

이 예제는 VPC vpc-88888888에서 인스턴스 i-0598c7d356eba48d7의 연결을 해제합니다.

명령:

```
aws ec2 detach-classic-link-vpc --instance-id i-0598c7d356eba48d7 --vpc-
id vpc-88888888
```

출력:

```
{
  "Return": true
}
```

- 자세한 API 내용은 명령 참조DetachClassicLinkVpc의 섹션을 참조하세요. AWS CLI

### `detach-internet-gateway`

다음 코드 예시에서는 `detach-internet-gateway`을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 인터넷 게이트웨이를 분리하려면 VPC

다음 `detach-internet-gateway` 예제에서는 지정된 인터넷 게이트웨이를 특정 에서 분리합니다VPC.

```
aws ec2 detach-internet-gateway \
    --internet-gateway-id igw-0d0fb496b3EXAMPLE \
    --vpc-id vpc-0a60eb65b4EXAMPLE
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon VPC 사용 설명서의 인터넷 게이트웨이를 참조하세요.

- 자세한 API 내용은 명령 참조DetachInternetGateway의 섹션을 참조하세요. AWS CLI

### detach-network-interface

다음 코드 예시에서는 detach-network-interface을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스에서 네트워크 인터페이스를 분리하려면

이 예제에서는 지정된 인스턴스에서 지정된 네트워크 인터페이스를 분리합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 detach-network-interface --attachment-id eni-attach-66c4350a
```

- 자세한 API 내용은 명령 참조DetachNetworkInterface의 섹션을 참조하세요. AWS CLI

### detach-verified-access-trust-provider

다음 코드 예시에서는 detach-verified-access-trust-provider을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스에서 신뢰 공급자를 분리하려면

다음 detach-verified-access-trust-provider 예제에서는 지정된 Verified Access 신뢰 공급자를 지정된 Verified Access 인스턴스에서 분리합니다.

```
aws ec2 detach-verified-access-trust-provider \
```

```
    --verified-access-instance-id vai-0ce000c0b7643abea \
    --verified-access-trust-provider-id vatp-0bb32de759a3e19e7
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Verified Access 사용 설명서의 Verified Access 인스턴스를 참조하세요. AWS

- 자세한 API 내용은 명령 참조DetachVerifiedAccessTrustProvider의 섹션을 참조하세요. AWS CLI

### `detach-volume`

다음 코드 예시에서는 `detach-volume`을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스에서 볼륨을 분리하려면

이 예제 명령은 연결된 인스턴스에서 볼륨(`vol-049df61146c4d7901`)을 분리합니다.

명령:

```
aws ec2 detach-volume --volume-id vol-1234567890abcdef0
```

출력:

```
{
    "AttachTime": "2014-02-27T19:23:06.000Z",
    "InstanceId": "i-1234567890abcdef0",
    "VolumeId": "vol-049df61146c4d7901",
    "State": "detaching",
    "Device": "/dev/sdb"
}
```

- 자세한 API 내용은 명령 참조DetachVolume의 섹션을 참조하세요. AWS CLI

### `detach-vpn-gateway`

다음 코드 예시에서는 `detach-vpn-gateway`을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 가상 프라이빗 게이트웨이를 분리하려면 VPC

이 예제에서는 지정된 가상 프라이빗 게이트웨이를 지정된 에서 분리합니다VPC. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 detach-vpn-gateway --vpn-gateway-id vgw-9a4cacf3 --vpc-id vpc-a01106c2
```

- 자세한 API 내용은 명령 참조DetachVpnGateway의 섹션을 참조하세요. AWS CLI

### `disable-address-transfer`

다음 코드 예시에서는 `disable-address-transfer`을 사용하는 방법을 보여 줍니다.

AWS CLI

탄력적 IP 주소 전송을 비활성화하려면

다음 `disable-address-transfer` 예제에서는 지정된 탄력적 IP 주소에 대한 탄력적 IP 주소 전송을 비활성화합니다.

```
aws ec2 disable-address-transfer \
```

```
    --allocation-id eipalloc-09ad461b0d03f6aaf
```

출력:

```
{
    "AddressTransfer": {
        "PublicIp": "100.21.184.216",
        "AllocationId": "eipalloc-09ad461b0d03f6aaf",
        "AddressTransferStatus": "disabled"
    }
}
```

자세한 내용은 Amazon VPC 사용 설명서의 탄력적 IP 주소 전송을 참조하세요.

- 자세한 API 내용은 명령 참조DisableAddressTransfer의 섹션을 참조하세요. AWS CLI

### disable-aws-network-performance-metric-subscription

다음 코드 예시에서는 disable-aws-network-performance-metric-subscription을 사용하는 방법을 보여 줍니다.

AWS CLI

지표 구독을 비활성화하려면

다음 disable-aws-network-performance-metric-subscription 예제에서는 지정된 소스 리전과 대상 리전 간의 집계 네트워크 지연 시간 모니터링을 비활성화합니다.

```
aws ec2 disable-aws-network-performance-metric-subscription \
    --source us-east-1 \
    --destination eu-west-1 \
    --metric aggregate-latency \
    --statistic p50
```

출력:

```
{
    "Output": true
}
```

자세한 내용은 인프라 성능 사용 설명서의 구독 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DisableAwsNetworkPerformanceMetricSubscription의 섹션을 참조하세요. AWS CLI

### disable-ebs-encryption-by-default

다음 코드 예시에서는 disable-ebs-encryption-by-default을 사용하는 방법을 보여 줍니다.

AWS CLI

기본적으로 EBS 암호화를 비활성화하려면

다음 disable-ebs-encryption-by-default 예제에서는 현재 리전의 AWS 계정에 대해 기본적으로 EBS 암호화를 비활성화합니다.

```
aws ec2 disable-ebs-encryption-by-default
```

출력:

```
{
    "EbsEncryptionByDefault": false
}
```

- 자세한 API 내용은 명령 참조DisableEbsEncryptionByDefault의 섹션을 참조하세요. AWS CLI

### disable-fast-launch

다음 코드 예시에서는 disable-fast-launch을 사용하는 방법을 보여 줍니다.

AWS CLI

이미지의 빠른 시작을 중단하려면

다음 disable-fast-launch 예제에서는 지정된 에서 빠른 시작을 중단하고 미리 프로비저닝된 기존 스냅샷을 AMI정리합니다.

```
aws ec2 disable-fast-launch \
    --image-id ami-01234567890abcedf
```

출력:

```
{
    "ImageId": "ami-01234567890abcedf",
    "ResourceType": "snapshot",
    "SnapshotConfiguration": {},
    "LaunchTemplate": {
        "LaunchTemplateId": "lt-01234567890abcedf",
        "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"
}
```

더 빠른 시작을 AMI 위해 Windows를 구성하는 방법에 대한 자세한 내용은 Amazon EC2 사용 설명서 의 더 빠른 시작을 AMI 위해 를 구성하는 단원을 참조하세요.

- 자세한 API 내용은 명령 참조DisableFastLaunch의 섹션을 참조하세요. AWS CLI

### disable-fast-snapshot-restores

다음 코드 예시에서는 disable-fast-snapshot-restores을 사용하는 방법을 보여 줍니다.

AWS CLI

빠른 스냅샷 복원을 비활성화하려면

다음 disable-fast-snapshot-restores 예제에서는 지정된 가용 영역에서 지정된 스냅샷에 대한 빠른 스냅샷 복원을 비활성화합니다.

```
aws ec2 disable-fast-snapshot-restores \
    --availability-zones us-east-2a \
    --source-snapshot-ids snap-1234567890abcdef0
```

출력:

```
{
    "Successful": [
```

```
        {
            "SnapshotId": "snap-1234567890abcdef0"
            "AvailabilityZone": "us-east-2a",
            "State": "disabling",
            "StateTransitionReason": "Client.UserInitiated",
            "OwnerId": "123456789012",
            "EnablingTime": "2020-01-25T23:57:49.602Z"
        }
    ],
    "Unsuccessful": []
}
```

- 자세한 API 내용은 명령 참조DisableFastSnapshotRestores의 섹션을 참조하세요. AWS CLI

### `disable-image-block-public-access`

다음 코드 예시에서는 `disable-image-block-public-access`을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 리전AMIs에서 에 대한 퍼블릭 액세스 차단을 비활성화하려면

다음 `disable-image-block-public-access` 예제에서는 지정된 리전AMIs의 계정 수준에서 에 대한 퍼블릭 액세스 차단을 비활성화합니다.

```
aws ec2 disable-image-block-public-access \
    --region us-east-1
```

출력:

```
{
    "ImageBlockPublicAccessState": "unblocked"
}
```

자세한 내용은 Amazon EC2 사용 설명서의 에 대한 퍼블릭 액세스 차단AMIs을 참조하세요.

- 자세한 API 내용은 명령 참조DisableImageBlockPublicAccess의 섹션을 참조하세요. AWS CLI

### `disable-image-deprecation`

다음 코드 예시에서는 `disable-image-deprecation`을 사용하는 방법을 보여 줍니다.

AWS CLI

의 사용 중지를 취소하려면 AMI

다음 disable-image-deprecation 예제에서는 describe-images 출력에서 DeprecationTime 필드를 AMI제거하는 의 사용 중지를 취소합니다. 이 절차를 수행하려면 AMI 소유자여야 합니다.

```
aws ec2 disable-image-deprecation \
    --image-id ami-1234567890abcdef0
```

출력:

```
{
    "RequestID": "11aabb229-4eac-35bd-99ed-be587EXAMPLE",
    "Return": "true"
}
```

자세한 내용은 Amazon EC2 사용 설명서의 AMI <https://docs.aws.amazon.com/AWS EC2/latest/UserGuide/ami-deprecate.html#deprecate-ami> 사용 중지를 참조하세요.

- 자세한 API 내용은 명령 참조DisableImageDeprecation의 섹션을 참조하세요. AWS CLI

### disable-image

다음 코드 예시에서는 disable-image을 사용하는 방법을 보여 줍니다.

AWS CLI

를 비활성화하려면 AMI

다음 disable-image 예제에서는 지정된 를 비활성화합니다AMI.

```
aws ec2 disable-image \
    --image-id ami-1234567890abcdef0
```

출력:

```
{
```

```
    "Return": "true"
}
```

자세한 내용은 Amazon EC2 사용 설명서의 비활성화AMI를 참조하세요.

- 자세한 API 내용은 명령 참조DisableImage의 섹션을 참조하세요. AWS CLI

### `disable-ipam-organization-admin-account`

다음 코드 예시에서는 `disable-ipam-organization-admin-account`을 사용하는 방법을 보여 줍니다.

AWS CLI

위임된 IPAM 관리자를 비활성화하려면

특정 시나리오에서는 AWS OrganizationsIPAM와 통합됩니다. 이렇게 하면 AWS Organizations 관리 계정이 AWS Organizations 멤버 계정을 IPAM 관리자로 위임합니다.

이 예에서는 IPAM 관리자 계정을 위임한 AWS Organizations 관리 계정이며 해당 계정이 IPAM 관리자가 되지 않도록 하려는 경우

이 요청을 할 `--region` 때 에 대해 모든 AWS 리전을 사용할 수 있습니다. 원래 관리자를 위임한 리전, 이 IPAM 생성된 리전 또는 IPAM 운영 리전을 사용할 필요가 없습니다. 위임된 관리자 계정을 비활성화하는 경우 언제든지 다시 활성화하거나 새 계정을 IPAM 관리자로 위임할 수 있습니다.

다음 `disable-ipam-organization-admin-account` 예제에서는 AWS 계정의 위임된 IPAM 관리자를 비활성화합니다.

```
aws ec2 disable-ipam-organization-admin-account \
    --delegated-admin-account-id 320805250157 \
    --region ap-south-1
```

출력:

```
{
    "Success": true
}
```

자세한 내용은 Amazon VPC IPAM 사용 설명서 의 AWS 조직 내 계정IPAM과 통합을 참조하세요.

- 자세한 API 내용은 명령 참조DisableIpamOrganizationAdminAccount의 섹션을 참조하세요. AWS CLI

### disable-serial-console-access

다음 코드 예시에서는 disable-serial-console-access을 사용하는 방법을 보여 줍니다.

AWS CLI

계정의 EC2 직렬 콘솔에 대한 액세스를 비활성화하려면

다음 disable-serial-console-access 예제에서는 직렬 콘솔에 대한 계정 액세스를 비활성화합니다.

```
aws ec2 disable-serial-console-access
```

출력:

```
{
    "SerialConsoleAccessEnabled": false
}
```

자세한 내용은 Amazon EC2 사용 설명서의 EC2 직렬 콘솔을 참조하세요.

- 자세한 API 내용은 명령 참조DisableSerialConsoleAccess의 섹션을 참조하세요. AWS CLI

### disable-snapshot-block-public-access

다음 코드 예시에서는 disable-snapshot-block-public-access을 사용하는 방법을 보여 줍니다.

AWS CLI

스냅샷에 대한 퍼블릭 액세스 차단을 비활성화하려면

다음 disable-snapshot-block-public-access 예제에서는 스냅샷의 퍼블릭 공유를 허용하기 위해 스냅샷에 대한 퍼블릭 액세스 차단을 비활성화합니다.

```
aws ec2 disable-snapshot-block-public-access
```

출력:

```
{
    "State": "unblocked"
}
```

자세한 내용은 Amazon EBS 사용 설명서의 스냅샷에 대한 퍼블릭 액세스 차단을 참조하세요.

- 자세한 API 내용은 명령 참조DisableSnapshotBlockPublicAccess의 섹션을 참조하세요. AWS CLI

### `disable-transit-gateway-route-table-propagation`

다음 코드 예시에서는 `disable-transit-gateway-route-table-propagation`을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 연결을 비활성화하여 지정된 전파 경로 테이블에 경로를 전파하려면

다음 `disable-transit-gateway-route-table-propagation` 예제에서는 지정된 연결을 비활성화하여 지정된 전파 라우팅 테이블에 경로를 전파합니다.

```
aws ec2 disable-transit-gateway-route-table-propagation \
    --transit-gateway-route-table-id tgw-rtb-0a823edbdeEXAMPLE \
    --transit-gateway-attachment-id tgw-attach-09b52ccdb5EXAMPLE
```

출력:

```
{
    "Propagation": {
        "TransitGatewayAttachmentId": "tgw-attach-09b52ccdb5EXAMPLE",
        "ResourceId": "vpc-4d7de228",
        "ResourceType": "vpc",
        "TransitGatewayRouteTableId": "tgw-rtb-0a823edbdeEXAMPLE",
        "State": "disabled"
    }
}
```

자세한 내용은 Transit Gateways Guide의 Transit Gateway route table을 참조하세요.

- 자세한 API 내용은 명령 참조DisableTransitGatewayRouteTablePropagation의 섹션을 참조하세요. AWS CLI

### `disable-vgw-route-propagation`

다음 코드 예시에서는 `disable-vgw-route-propagation`을 사용하는 방법을 보여 줍니다.

AWS CLI

라우팅 전파를 비활성화하려면

이 예제에서는 지정된 가상 프라이빗 게이트웨이가 정적 경로를 지정된 라우팅 테이블에 전파하지 못하도록 비활성화합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 disable-vgw-route-propagation --route-table-id rtb-22574640 --gateway-
id vgw-9a4cacf3
```

- 자세한 API 내용은 명령 참조DisableVgwRoutePropagation의 섹션을 참조하세요. AWS CLI

### `disable-vpc-classic-link-dns-support`

다음 코드 예시에서는 `disable-vpc-classic-link-dns-support`을 사용하는 방법을 보여 줍니다.

AWS CLI

에 대한 지원을 비활성화 ClassicLink DNS하려면 VPC

이 예제에서는 에 대한 지원을 비활성화합니다 ClassicLink DNSvpc-88888888.

명령:

```
aws ec2 disable-vpc-classic-link-dns-support --vpc-id vpc-88888888
```

출력:

```
{
  "Return": true
}
```

- 자세한 API 내용은 명령 참조DisableVpcClassicLinkDnsSupport의 섹션을 참조하세요. AWS CLI

### `disable-vpc-classic-link`

다음 코드 예시에서는 `disable-vpc-classic-link`을 사용하는 방법을 보여 줍니다.

AWS CLI

에 ClassicLink 대해 를 비활성화하려면 VPC

이 예제에서는 vpc-8888888 ClassicLink 에 대해 를 비활성화합니다.

명령:

```
aws ec2 disable-vpc-classic-link --vpc-id vpc-88888888
```

출력:

```
{
  "Return": true
}
```

- 자세한 API 내용은 명령 참조DisableVpcClassicLink의 섹션을 참조하세요. AWS CLI

### `disassociate-address`

다음 코드 예시에서는 `disassociate-address`을 사용하는 방법을 보여 줍니다.

AWS CLI

EC2-Classic에서 탄력적 IP 주소 연결을 해제하려면

이 예제에서는 EC2-Classic의 인스턴스에서 탄력적 IP 주소의 연결을 해제합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 disassociate-address --public-ip 198.51.100.0
```

에서 탄력적 IP 주소 연결을 해제하려면 EC2VPC

이 예제에서는 의 인스턴스에서 탄력적 IP 주소의 연결을 해제합니다VPC. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 disassociate-address --association-id eipassoc-2bebb745
```

- 자세한 API 내용은 명령 참조DisassociateAddress의 섹션을 참조하세요. AWS CLI

### disassociate-client-vpn-target-network

다음 코드 예시에서는 disassociate-client-vpn-target-network을 사용하는 방법을 보여 줍니다.

AWS CLI

클라이언트 VPN 엔드포인트에서 네트워크 연결을 해제하려면

다음 disassociate-client-vpn-target-network 예제에서는 지정된 클라이언트 VPN 엔드포인트의 cvpn-assoc-12312312312312312 연결 ID와 연결된 대상 네트워크의 연결을 해제합니다.

```
aws ec2 disassociate-client-vpn-target-network \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \
    --association-id cvpn-assoc-12312312312312312
```

출력:

```
{
    "AssociationId": "cvpn-assoc-12312312312312312",
    "Status": {
        "Code": "disassociating"
    }
}
```

자세한 내용은 AWS 클라이언트 VPN 관리자 안내서의 대상 네트워크를 참조하세요.

- 자세한 API 내용은 명령 참조DisassociateClientVpnTargetNetwork의 섹션을 참조하세요. AWS CLI

### disassociate-iam-instance-profile

다음 코드 예시에서는 disassociate-iam-instance-profile을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 인스턴스 프로파일 연결을 해제하려면

이 예제에서는 IAM 인스턴스 프로파일을 연결 ID 와 연결 해제합니다`iip-assoc-05020b59952902f5f`.

명령:

```
aws ec2 disassociate-iam-instance-profile --association-id iip-assoc-05020b59952902f5f
```

출력:

```
{
  "IamInstanceProfileAssociation": {
      "InstanceId": "i-123456789abcde123",
      "State": "disassociating",
      "AssociationId": "iip-assoc-05020b59952902f5f",
      "IamInstanceProfile": {
          "Id": "AIPAI5IVIHMFFYY2DKV5Y",
          "Arn": "arn:aws:iam::123456789012:instance-profile/admin-role"
      }
  }
}
```

- 자세한 API 내용은 명령 참조DisassociateIamInstanceProfile의 섹션을 참조하세요. AWS CLI

### `disassociate-instance-event-window`

다음 코드 예시에서는 `disassociate-instance-event-window`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 이벤트 창에서 하나 이상의 인스턴스 연결을 해제하려면

다음 `disassociate-instance-event-window` 예제에서는 이벤트 창에서 하나 이상의 인스턴스를 연결 해제합니다. `instance-event-window-id` 파라미터를 지정하여 이벤트 창을 지정합니다. 인스턴스 연결을 해제하려면 `association-target` 파라미터를 지정하고 파라미터 값에는 하나 이상의 인스턴스를 지정합니다IDs.

```
aws ec2 disassociate-instance-event-window \
    --region us-east-1 \
    --instance-event-window-id iew-0abcdef1234567890 \
    --association-target "InstanceIds=i-1234567890abcdef0,i-0598c7d356eba48d7"
```

출력:

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating"
    }
}
```

이벤트 기간 제약 조건은 Amazon EC2 사용 설명서의 예약된 이벤트 섹션의 고려 사항을 참조하세요.

예제 2: 이벤트 창에서 인스턴스 태그 연결을 해제하려면

다음 `disassociate-instance-event-window` 예제에서는 이벤트 창에서 인스턴스 태그를 연결 해제합니다. `instance-event-window-id` 파라미터를 지정하여 이벤트 창을 지정합니다. 인스턴스 태그를 연결 해제하려면 `association-target` 파라미터를 지정하고 파라미터 값으로 하나 이상의 태그를 지정합니다.

```
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}]"
```

출력:

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
```

```
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating"
    }
}
```

이벤트 기간 제약 조건은 Amazon EC2 사용 설명서의 예약된 이벤트 섹션의 고려 사항을 참조하세요.

예제 3: 이벤트 창에서 전용 호스트 연결을 해제하려면

다음 `disassociate-instance-event-window` 예제에서는 이벤트 창에서 전용 호스트의 연결을 해제합니다. `instance-event-window-id` 파라미터를 지정하여 이벤트 창을 지정합니다. 전용 호스트의 연결을 해제하려면 `association-target` 파라미터를 지정하고 파라미터 값에는 하나 이상의 전용 호스트 를 지정합니다IDs.

```
aws ec2 disassociate-instance-event-window \
    --region us-east-1 \
    --instance-event-window-id iew-0abcdef1234567890 \
    --association-target DedicatedHostIds=h-029fa35a02b99801d
```

출력:

```
{
    "InstanceEventWindow": {
        "InstanceEventWindowId": "iew-0abcdef1234567890",
        "Name": "myEventWindowName",
        "CronExpression": "* 21-23 * * 2,3",
        "AssociationTarget": {
            "InstanceIds": [],
            "Tags": [],
            "DedicatedHostIds": []
        },
        "State": "creating"
    }
}
```

이벤트 기간 제약 조건은 Amazon EC2 사용 설명서의 예약된 이벤트 섹션의 고려 사항을 참조하세요.

- 자세한 API 내용은 명령 참조DisassociateInstanceEventWindow의 섹션을 참조하세요. AWS CLI

### `disassociate-ipam-resource-discovery`

다음 코드 예시에서는 `disassociate-ipam-resource-discovery`을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 리소스 검색을 연결 해제하려면 IPAM

이 예제에서는 IPAM 위임된 관리자 계정이며 에서 IPAM 리소스 검색을 연결 해제하려고 합니다 IPAM. 설명 명령을 실행한 후 `"ResourceDiscoveryStatus": "not-found"` 와 의 연결을 해제하여 다른 연결을 위한 공간을 IPAM 만들고 싶다는 사실을 알게 되었습니다.

다음 `disassociate-ipam-resource-discovery` 예제에서는 AWS 계정의 IPAM 리소스 검색 연결을 해제합니다.

```
aws ec2 disassociate-ipam-resource-discovery \
    --ipam-resource-discovery-association-id ipam-res-disco-assoc-04382a6346357cf82
 \
    --region us-east-1
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon VPC IPAM 사용 설명서의 조직 외부 계정IPAM과 통합을 참조하세요.

- 자세한 API 내용은 명령 참조DisassociateIpamResourceDiscovery의 섹션을 참조하세요. AWS CLI

### `disassociate-nat-gateway-address`

다음 코드 예시에서는 `disassociate-nat-gateway-address`을 사용하는 방법을 보여 줍니다.

AWS CLI

퍼블릭 NAT 게이트웨이에서 탄력적 IP 주소 연결을 해제하려면

다음 `disassociate-nat-gateway-address` 예제에서는 지정된 퍼블릭 NAT 게이트웨이에서 지정된 탄력적 IP 주소의 연결을 해제합니다.

```
aws ec2 disassociate-nat-gateway-address \
    --nat-gateway-id nat-1234567890abcdef0 \
    --association-ids eipassoc-0f96bdca17EXAMPLE
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon VPC 사용 설명서의 NAT 게이트웨이를 참조하세요.

- 자세한 API 내용은 명령 참조DisassociateNatGatewayAddress의 섹션을 참조하세요. AWS CLI

### disassociate-route-table

다음 코드 예시에서는 disassociate-route-table을 사용하는 방법을 보여 줍니다.

AWS CLI

라우팅 테이블 연결을 해제하려면

이 예제에서는 지정된 서브넷에서 지정된 라우팅 테이블의 연결을 해제합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 disassociate-route-table --association-id rtbassoc-781d0d1a
```

- 자세한 API 내용은 명령 참조DisassociateRouteTable의 섹션을 참조하세요. AWS CLI

### disassociate-subnet-cidr-block

다음 코드 예시에서는 disassociate-subnet-cidr-block을 사용하는 방법을 보여 줍니다.

AWS CLI

서브넷에서 IPv6 CIDR 블록 연결을 해제하려면

이 예제에서는 IPv6 CIDR 블록의 연결 ID를 사용하여 서브넷에서 CIDR 블록의 연결을 해제합니다.

명령:

```
aws ec2 disassociate-subnet-cidr-block --association-id subnet-cidr-assoc-3aa54053
```

출력:

```
{
  "SubnetId": "subnet-5f46ec3b",
  "Ipv6CidrBlockAssociation": {
      "Ipv6CidrBlock": "2001:db8:1234:1a00::/64",
      "AssociationId": "subnet-cidr-assoc-3aa54053",
```

```
        "Ipv6CidrBlockState": {
            "State": "disassociating"
        }
    }
}
```

- 자세한 API 내용은 명령 참조DisassociateSubnetCidrBlock의 섹션을 참조하세요. AWS CLI

### `disassociate-transit-gateway-multicast-domain`

다음 코드 예시에서는 `disassociate-transit-gateway-multicast-domain`을 사용하는 방법을 보여 줍니다.

AWS CLI

멀티캐스트 도메인에서 서브넷 연결을 해제하려면

다음 `disassociate-transit-gateway-multicast-domain` 예제에서는 지정된 멀티캐스트 도메인에서 서브넷의 연결을 해제합니다.

```
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
```

출력:

```
{
    "Associations": {
        "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef7EXAMPLE",
        "TransitGatewayAttachmentId": "tgw-attach-070e571cd1EXAMPLE",
        "ResourceId": "vpc-7EXAMPLE",
        "ResourceType": "vpc",
        "Subnets": [
            {
                "SubnetId": "subnet-000de86e3bEXAMPLE",
                "State": "disassociating"
            }
        ]
    }
}
```

자세한 내용은 Transit Gateways Guide '의 멀티캐스트 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DisassociateTransitGatewayMulticastDomain의 섹션을 참조하세요. AWS CLI

### `disassociate-transit-gateway-route-table`

다음 코드 예시에서는 `disassociate-transit-gateway-route-table`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 연결에서 전송 게이트웨이 라우팅 테이블의 연결을 해제하려면

다음 `disassociate-transit-gateway-route-table` 예제에서는 전송 게이트웨이 라우팅 테이블에서 지정된 연결을 연결 해제합니다.

```
aws ec2 disassociate-transit-gateway-route-table \
    --transit-gateway-route-table-id tgw-rtb-002573ed1eEXAMPLE \
    --transit-gateway-attachment-id tgw-attach-08e0bc912cEXAMPLE
```

출력:

```
{
    "Association": {
        "TransitGatewayRouteTableId": "tgw-rtb-002573ed1eEXAMPLE",
        "TransitGatewayAttachmentId": "tgw-attach-08e0bc912cEXAMPLE",
        "ResourceId": "11460968-4ac1-4fd3-bdb2-00599EXAMPLE",
        "ResourceType": "direct-connect-gateway",
        "State": "disassociating"
    }
}
```

자세한 내용은 Transit Gateways Guide의 Transit Gateway route table을 참조하세요.

- 자세한 API 내용은 명령 참조DisassociateTransitGatewayRouteTable의 섹션을 참조하세요. AWS CLI

### `disassociate-vpc-cidr-block`

다음 코드 예시에서는 `disassociate-vpc-cidr-block`을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 IPv6 CIDR 블록을 연결 해제하려면 VPC

이 예제에서는 IPv6 CIDR 블록에 대한 연결 ID를 VPC 사용하여 에서 CIDR 블록을 연결 해제합니다.

명령:

```
aws ec2 disassociate-vpc-cidr-block --association-id vpc-cidr-assoc-eca54085
```

출력:

```
{
  "Ipv6CidrBlockAssociation": {
      "Ipv6CidrBlock": "2001:db8:1234:1a00::/56",
      "AssociationId": "vpc-cidr-assoc-eca54085",
      "Ipv6CidrBlockState": {
          "State": "disassociating"
      }
  },
  "VpcId": "vpc-a034d6c4"
}
```

에서 IPv4 CIDR 블록을 연결 해제하려면 VPC

이 예제는 에서 IPv4 CIDR 블록의 연결을 해제합니다VPC.

명령:

```
aws ec2 disassociate-vpc-cidr-block --association-id vpc-cidr-assoc-0287ac6b
```

출력:

```
{
  "CidrBlockAssociation": {
      "AssociationId": "vpc-cidr-assoc-0287ac6b",
      "CidrBlock": "172.18.0.0/16",
      "CidrBlockState": {
          "State": "disassociating"
      }
```

```
  },
  "VpcId": "vpc-27621243"
}
```

- 자세한 API 내용은 명령 참조DisassociateVpcCidrBlock의 섹션을 참조하세요. AWS CLI

### enable-address-transfer

다음 코드 예시에서는 enable-address-transfer을 사용하는 방법을 보여 줍니다.

AWS CLI

탄력적 IP 주소 전송을 활성화하려면

다음 enable-address-transfer 예제에서는 지정된 탄력적 IP 주소에 대해 탄력적 IP 주소를 지정된 계정으로 전송할 수 있습니다.

```
aws ec2 enable-address-transfer \
    --allocation-id eipalloc-09ad461b0d03f6aaf \
    --transfer-account-id 123456789012
```

출력:

```
{
     "AddressTransfer": {
        "PublicIp": "100.21.184.216",
        "AllocationId": "eipalloc-09ad461b0d03f6aaf",
        "TransferAccountId": "123456789012",
        "TransferOfferExpirationTimestamp": "2023-02-22T20:51:01.000Z",
        "AddressTransferStatus": "pending"
    }
}
```

자세한 내용은 Amazon VPC 사용 설명서의 탄력적 IP 주소 전송을 참조하세요.

- 자세한 API 내용은 명령 참조EnableAddressTransfer의 섹션을 참조하세요. AWS CLI

### enable-aws-network-performance-metric-subscription

다음 코드 예시에서는 enable-aws-network-performance-metric-subscription을 사용하는 방법을 보여 줍니다.

AWS CLI

지표 구독을 활성화하려면

다음 enable-aws-network-performance-metric-subscription 예제에서는 지정된 소스 리전과 대상 리전 간의 집계 네트워크 지연 시간을 모니터링할 수 있습니다.

```
aws ec2 enable-aws-network-performance-metric-subscription \
    --source us-east-1 \
    --destination eu-west-1 \
    --metric aggregate-latency \
    --statistic p50
```

출력:

```
{
    "Output": true
}
```

자세한 내용은 인프라 성능 사용 설명서의 구독 관리를 참조하세요.

- 자세한 API 내용은 명령 참조EnableAwsNetworkPerformanceMetricSubscription의 섹션을 참조하세요. AWS CLI

### enable-ebs-encryption-by-default

다음 코드 예시에서는 enable-ebs-encryption-by-default을 사용하는 방법을 보여 줍니다.

AWS CLI

기본적으로 EBS 암호화를 활성화하려면

다음 enable-ebs-encryption-by-default 예제에서는 현재 리전의 AWS 계정에 대해 기본적으로 EBS 암호화를 활성화합니다.

```
aws ec2 enable-ebs-encryption-by-default
```

출력:

```
{
```

```
    "EbsEncryptionByDefault": true
}
```

- 자세한 API 내용은 명령 참조EnableEbsEncryptionByDefault의 섹션을 참조하세요. AWS CLI

### enable-fast-launch

다음 코드 예시에서는 enable-fast-launch을 사용하는 방법을 보여 줍니다.

AWS CLI

이미지의 빠른 시작

다음 enable-fast-launch 예제에서는 지정된 에서 빠른 시작을 시작하고 시작할 최대 병렬 인스턴스 수를 6으로 AMI 설정합니다. 를 사전 프로비저닝하는 데 사용할 리소스 유형은 기본값snapshot인 로 AMI 설정됩니다.

```
aws ec2 enable-fast-launch \
    --image-id ami-01234567890abcedf \
    --max-parallel-launches 6 \
    --resource-type snapshot
```

출력:

```
{
    "ImageId": "ami-01234567890abcedf",
    "ResourceType": "snapshot",
    "SnapshotConfiguration": {
        "TargetResourceCount": 10
    },
    "LaunchTemplate": {},
    "MaxParallelLaunches": 6,
    "OwnerId": "0123456789123",
    "State": "enabling",
    "StateTransitionReason": "Client.UserInitiated",
    "StateTransitionTime": "2022-01-27T22:16:03.199000+00:00"
}
```

더 빠른 시작을 AMI 위해 Windows를 구성하는 방법에 대한 자세한 내용은 Amazon EC2 사용 설명서 의 더 빠른 시작을 AMI 위해 를 구성하는 단원을 참조하세요.

- 자세한 API 내용은 명령 참조EnableFastLaunch의 섹션을 참조하세요. AWS CLI

### enable-fast-snapshot-restores

다음 코드 예시에서는 enable-fast-snapshot-restores을 사용하는 방법을 보여 줍니다.

AWS CLI

빠른 스냅샷 복원을 활성화하려면

다음 enable-fast-snapshot-restores 예제에서는 지정된 가용 영역에서 지정된 스냅샷에 대해 빠른 스냅샷 복원을 활성화합니다.

```
aws ec2 enable-fast-snapshot-restores \
    --availability-zones us-east-2a us-east-2b \
    --source-snapshot-ids snap-1234567890abcdef0
```

출력:

```
{
    "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": []
}
```

- 자세한 API 내용은 명령 참조EnableFastSnapshotRestores의 섹션을 참조하세요. AWS CLI

### enable-image-block-public-access

다음 코드 예시에서는 enable-image-block-public-access을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 리전AMIs에서 에 대한 퍼블릭 액세스 차단을 활성화하려면

다음 enable-image-block-public-access 예제에서는 지정된 리전AMIs의 계정 수준에서 에 대한 퍼블릭 액세스 차단을 활성화합니다.

```
aws ec2 enable-image-block-public-access \
    --region us-east-1 \
    --image-block-public-access-state block-new-sharing
```

출력:

```
{
    "ImageBlockPublicAccessState": "block-new-sharing"
}
```

자세한 내용은 Amazon EC2 사용 설명서의 에 대한 퍼블릭 액세스 차단AMIs을 참조하세요.

- 자세한 API 내용은 명령 참조EnableImageBlockPublicAccess의 섹션을 참조하세요. AWS CLI

### enable-image-deprecation

다음 코드 예시에서는 enable-image-deprecation을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 의 사용을 중단하려면 AMI

다음 enable-image-deprecation 예제에서는 특정 날짜 및 시간에 AMI 의 사용을 중단합니다. 초 단위로 값을 지정하면 Amazon은 초를 가장 가까운 분으로 EC2 반올림합니다. 이 절차를 수행하려면 AMI 소유자여야 합니다.

```
aws ec2 enable-image-deprecation \
    --image-id ami-1234567890abcdef0 \
    --deprecate-at "2022-10-15T13:17:12.000Z"
```

출력:

```
{
    "RequestID": "59dbff89-35bd-4eac-99ed-be587EXAMPLE",
    "Return": "true"
}
```

자세한 내용은 Amazon EC2 사용 설명서의 AMI <https://docs.aws.amazon.com/AWS EC2/latest/UserGuide/ami-deprecate.html#deprecate-ami> 사용 중지를 참조하세요.

- 자세한 API 내용은 명령 참조EnableImageDeprecation의 섹션을 참조하세요. AWS CLI

### `enable-image`

다음 코드 예시에서는 `enable-image`을 사용하는 방법을 보여 줍니다.

AWS CLI

를 활성화하려면 AMI

다음 `enable-image` 예제에서는 지정된 를 활성화합니다AMI.

```
aws ec2 enable-image \
    --image-id ami-1234567890abcdef0
```

출력:

```
{
    "Return": "true"
}
```

자세한 내용은 Amazon EC2 사용 설명서의 비활성화AMI를 참조하세요.

- 자세한 API 내용은 명령 참조EnableImage의 섹션을 참조하세요. AWS CLI

### `enable-ipam-organization-admin-account`

다음 코드 예시에서는 `enable-ipam-organization-admin-account`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS Organizations와 통합하고 멤버 계정을 IPAM 계정으로 위임하려면

다음 enable-ipam-organization-admin-account 예제는 AWS OrganizationsIPAM와 통합되고 멤버 계정을 IPAM 계정으로 위임합니다.

```
aws ec2 enable-ipam-organization-admin-account \
    --delegated-admin-account-id 320805250157
```

출력:

```
{
    "Success": true
}
```

자세한 내용은 Amazon VPC IPAM 사용 설명서의 AWS 조직IPAM과 통합을 참조하세요.

- 자세한 API 내용은 명령 참조EnableIpamOrganizationAdminAccount의 섹션을 참조하세요. AWS CLI

### enable-reachability-analyzer-organization-sharing

다음 코드 예시에서는 enable-reachability-analyzer-organization-sharing을 사용하는 방법을 보여 줍니다.

AWS CLI

Reachability Analyzer에 대한 신뢰할 수 있는 액세스를 활성화하려면

다음 enable-reachability-analyzer-organization-sharing 예제에서는 Reachability Analyzer에 대한 신뢰할 수 있는 액세스를 활성화합니다.

```
aws ec2 enable-reachability-analyzer-organization-sharing
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Reachability Analyzer 사용 설명서의 교차 계정 분석을 참조하세요.

- 자세한 API 내용은 명령 참조EnableReachabilityAnalyzerOrganizationSharing의 섹션을 참조하세요. AWS CLI

### enable-serial-console-access

다음 코드 예시에서는 enable-serial-console-access을 사용하는 방법을 보여 줍니다.

AWS CLI

계정의 직렬 콘솔에 대한 액세스를 활성화하려면

다음 enable-serial-console-access 예제에서는 직렬 콘솔에 대한 계정 액세스를 활성화합니다.

```
aws ec2 enable-serial-console-access
```

출력:

```
{
    "SerialConsoleAccessEnabled": true
}
```

자세한 내용은 Amazon EC2 사용 설명서의 EC2 직렬 콘솔을 참조하세요.

- 자세한 API 내용은 명령 참조EnableSerialConsoleAccess의 섹션을 참조하세요. AWS CLI

### enable-snapshot-block-public-access

다음 코드 예시에서는 enable-snapshot-block-public-access을 사용하는 방법을 보여 줍니다.

AWS CLI

스냅샷에 대한 퍼블릭 액세스 차단을 활성화하려면

다음 enable-snapshot-block-public-access 예제에서는 스냅샷의 모든 퍼블릭 공유를 차단합니다.

```
aws ec2 enable-snapshot-block-public-access \
    --state block-all-sharing
```

출력:

```
{
    "State": "block-all-sharing"
}
```

자세한 내용은 Amazon EBS 사용 설명서의 스냅샷에 대한 퍼블릭 액세스 차단을 참조하세요.

- 자세한 API 내용은 명령 참조EnableSnapshotBlockPublicAccess의 섹션을 참조하세요. AWS CLI

### `enable-transit-gateway-route-table-propagation`

다음 코드 예시에서는 `enable-transit-gateway-route-table-propagation`을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 연결을 활성화하여 경로를 지정된 전파 라우팅 테이블에 전파하려면

다음 `enable-transit-gateway-route-table-propagation` 예제에서는 지정된 연결에서 지정된 전파 라우팅 테이블에 라우팅을 전파할 수 있습니다.

```
aws ec2 enable-transit-gateway-route-table-propagation \
    --transit-gateway-route-table-id tgw-rtb-0a823edbdeEXAMPLE \
    --transit-gateway-attachment-id tgw-attach-09b52ccdb5EXAMPLE
```

출력:

```
{
    "Propagation": {
        "TransitGatewayAttachmentId": "tgw-attach-09b52ccdb5EXAMPLE",
        "ResourceId": "vpc-4d7de228",
        "ResourceType": "vpc",
        "TransitGatewayRouteTableId": "tgw-rtb-0a823edbdeEXAMPLE",
        "State": "disabled"
    }
}
```

자세한 내용은 Transit Gateways Guide의 Transit Gateway route table을 참조하세요.

- 자세한 API 내용은 명령 참조EnableTransitGatewayRouteTablePropagation의 섹션을 참조하세요. AWS CLI

### `enable-vgw-route-propagation`

다음 코드 예시에서는 `enable-vgw-route-propagation`을 사용하는 방법을 보여 줍니다.

AWS CLI

라우팅 전파를 활성화하려면

이 예제에서는 지정된 가상 프라이빗 게이트웨이가 정적 경로를 지정된 라우팅 테이블에 전파할 수 있습니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 enable-vgw-route-propagation --route-table-id rtb-22574640 --gateway-
id vgw-9a4cacf3
```

- 자세한 API 내용은 명령 참조EnableVgwRoutePropagation의 섹션을 참조하세요. AWS CLI

### enable-volume-io

다음 코드 예시에서는 enable-volume-io을 사용하는 방법을 보여 줍니다.

AWS CLI

볼륨에 대해 I/O를 활성화하려면

이 예제에서는 볼륨 에서 I/O를 활성화합니다vol-1234567890abcdef0.

명령:

```
aws ec2 enable-volume-io --volume-id vol-1234567890abcdef0
```

출력:

```
{
  "Return": true
}
```

- 자세한 API 내용은 명령 참조EnableVolumeIo의 섹션을 참조하세요. AWS CLI

### enable-vpc-classic-link-dns-support

다음 코드 예시에서는 enable-vpc-classic-link-dns-support을 사용하는 방법을 보여 줍니다.

AWS CLI

에 대한 지원을 활성화 ClassicLink DNS하려면 VPC

이 예제에서는 에 대한 지원을 활성화합니다 ClassicLink DNSvpc-88888888.

명령:

```
aws ec2 enable-vpc-classic-link-dns-support --vpc-id vpc-88888888
```

출력:

```
{
  "Return": true
}
```

- 자세한 API 내용은 명령 참조EnableVpcClassicLinkDnsSupport의 섹션을 참조하세요. AWS CLI

### enable-vpc-classic-link

다음 코드 예시에서는 enable-vpc-classic-link을 사용하는 방법을 보여 줍니다.

AWS CLI

에 VPC 대해 를 활성화하려면 ClassicLink

이 예제에서는 에 대해 vpc-8888888를 활성화합니다 ClassicLink.

명령:

```
aws ec2 enable-vpc-classic-link --vpc-id vpc-88888888
```

출력:

```
{
  "Return": true
}
```

- 자세한 API 내용은 명령 참조EnableVpcClassicLink의 섹션을 참조하세요. AWS CLI

### export-client-vpn-client-certificate-revocation-list

다음 코드 예시에서는 export-client-vpn-client-certificate-revocation-list을 사용하는 방법을 보여 줍니다.

AWS CLI

클라이언트 인증서 취소 목록을 내보내려면

다음 export-client-vpn-client-certificate-revocation-list 예제에서는 지정된 클라이언트 VPN 엔드포인트에 대한 클라이언트 인증서 취소 목록을 내보냅니다. 이 예제에서는 읽기 쉽도록 출력이 텍스트 형식으로 반환됩니다.

```
aws ec2 export-client-vpn-client-certificate-revocation-list \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \
    --output text
```

출력:

```
-----BEGIN X509 CRL-----
MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END X509 CRL-----
STATUS      pending
```

자세한 내용은 클라이언트 관리자 안내서의 클라이언트 인증서 취소 목록을 참조하세요. AWS VPN

- 자세한 API 내용은 명령 참조ExportClientVpnClientCertificateRevocationList의 섹션을 참조하세요. AWS CLI

### export-client-vpn-client-configuration

다음 코드 예시에서는 export-client-vpn-client-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

클라이언트 구성을 내보내려면

다음 export-client-vpn-client-configuration 예제에서는 지정된 클라이언트 VPN 엔드포인트에 대한 클라이언트 구성을 내보냅니다. 이 예제에서는 읽기 쉽도록 출력이 텍스트 형식으로 반환됩니다.

```
aws ec2 export-client-vpn-client-configuration \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \
    --output text
```

출력:

```
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-----
MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
```

```
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----
</ca>
reneg-sec 0
```

자세한 내용은 클라이언트 관리자 안내서의 클라이언트 VPN 엔드포인트를 참조하세요. AWS VPN

- 자세한 API 내용은 명령 참조ExportClientVpnClientConfiguration의 섹션을 참조하세요. AWS CLI

### `export-image`

다음 코드 예시에서는 `export-image`을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 VM을 내보내려면 AMI

다음 `export-image` 예제에서는 지정된 형식으로 지정된 버킷AMI에 지정된 를 내보냅니다.

```
aws ec2 export-image \
    --image-id ami-1234567890abcdef0 \
    --disk-image-format VMDK \
    --s3-export-location S3Bucket=my-export-bucket,S3Prefix=exports/
```

출력:

```
{
    "DiskImageFormat": "vmdk",
    "ExportImageTaskId": "export-ami-1234567890abcdef0"
    "ImageId": "ami-1234567890abcdef0",
    "RoleName": "vmimport",
    "Progress": "0",
    "S3ExportLocation": {
        "S3Bucket": "my-export-bucket",
        "S3Prefix": "exports/"
    },
    "Status": "active",
    "StatusMessage": "validating"
```

```
}
```

- 자세한 API 내용은 명령 참조ExportImage의 섹션을 참조하세요. AWS CLI

### `get-associated-ipv6-pool-cidrs`

다음 코드 예시에서는 `get-associated-ipv6-pool-cidrs`을 사용하는 방법을 보여 줍니다.

AWS CLI

IPv6 주소 풀에 대한 연결을 가져오려면

다음 `get-associated-ipv6-pool-cidrs` 예제에서는 지정된 IPv6 주소 풀에 대한 연결을 가져옵니다.

```
aws ec2 get-associated-ipv6-pool-cidrs \
    --pool-id ipv6pool-ec2-012345abc12345abc
```

출력:

```
{
    "Ipv6CidrAssociations": [
        {
            "Ipv6Cidr": "2001:db8:1234:1a00::/56",
            "AssociatedResource": "vpc-111111222222333ab"
        }
    ]
}
```

- API 자세한 내용은 AWS CLI 명령 참조의 GetAssociatedIpv6PoolCidrs을 참조하세요.

### `get-aws-network-performance-data`

다음 코드 예시에서는 `get-aws-network-performance-data`을 사용하는 방법을 보여 줍니다.

AWS CLI

네트워크 성능 데이터를 가져오려면

다음 `get-aws-network-performance-data` 예제에서는 지정된 기간 동안 지정된 리전 간의 네트워크 성능에 대한 데이터를 검색합니다.

```
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
```

출력:


```
{
    "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"
                }
            ]
        }
    ]
}
```

자세한 내용은 인프라 성능 사용 설명서의 네트워크 성능 모니터링을 참조하세요.

- 자세한 API 내용은 명령 참조GetAwsNetworkPerformanceData의 섹션을 참조하세요. AWS CLI

### get-capacity-reservation-usage

다음 코드 예시에서는 get-capacity-reservation-usage을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 계정 간 용량 예약 사용량을 보려면

다음 get-capacity-reservation-usage 예제에서는 지정된 용량 예약에 대한 사용 정보를 표시합니다.

```
aws ec2 get-capacity-reservation-usage \
    --capacity-reservation-id cr-1234abcd56EXAMPLE
```

출력:

```
{
```

```
    "CapacityReservationId": "cr-1234abcd56EXAMPLE ",
    "InstanceUsages": [
        {
            "UsedInstanceCount": 1,
            "AccountId": "123456789012"
        }
    ],
    "AvailableInstanceCount": 4,
    "TotalInstanceCount": 5,
    "State": "active",
    "InstanceType": "t2.medium"
}
```

자세한 내용은 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 공유 용량 예약 사용량 보기를 참조하세요.

- 자세한 API 내용은 명령 참조GetCapacityReservationUsage의 섹션을 참조하세요. AWS CLI

### get-coip-pool-usage

다음 코드 예시에서는 get-coip-pool-usage을 사용하는 방법을 보여 줍니다.

AWS CLI

고객 소유 IP 주소 풀 사용을 가져오려면

다음 get-coip-pool-usage 예제에서는 지정된 고객 소유 IP 주소 풀의 사용 세부 정보를 가져옵니다.

```
aws ec2 get-coip-pool-usage \
    --pool-id ipv4pool-coip-123a45678bEXAMPLE
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS Outposts 사용 설명서의 고객 소유 IP 주소를 참조하세요.

- 자세한 API 내용은 명령 참조GetCoipPoolUsage의 섹션을 참조하세요. AWS CLI

### `get-console-output`

다음 코드 예시에서는 `get-console-output`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 콘솔 출력을 가져오려면

다음 `get-console-output` 예제에서는 지정된 Linux 인스턴스에 대한 콘솔 출력을 가져옵니다.

```
aws ec2 get-console-output \
    --instance-id i-1234567890abcdef0
```

출력:

```
{
    "InstanceId": "i-1234567890abcdef0",
    "Timestamp": "2013-07-25T21:23:53.000Z",
    "Output": "..."
}
```

자세한 내용은 Amazon EC2 사용 설명서의 인스턴스 콘솔 출력을 참조하세요.

예제 2: 최신 콘솔 출력을 가져오려면

다음 `get-console-output` 예제에서는 지정된 Linux 인스턴스에 대한 최신 콘솔 출력을 가져옵니다.

```
aws ec2 get-console-output \
    --instance-id i-1234567890abcdef0 \
    --latest \
    --output text
```

출력:

```
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
```

자세한 내용은 Amazon EC2 사용 설명서의 인스턴스 콘솔 출력을 참조하세요.

- 자세한 API 내용은 명령 참조GetConsoleOutput의 섹션을 참조하세요. AWS CLI

### get-console-screenshot

다음 코드 예시에서는 get-console-screenshot을 사용하는 방법을 보여 줍니다.

AWS CLI

실행 중인 인스턴스의 스크린샷을 검색하려면

다음 get-console-screenshot 예제에서는 지정된 인스턴스의 스크린샷을 .jpg 형식으로 검색합니다. 스크린샷은 Base64-encoded 문자열로 반환됩니다.

```
aws ec2 get-console-screenshot \
    --instance-id i-1234567890abcdef0
```

출력:

```
{
    "ImageData": "997987/8kgj49ikjhewkwwe0008084EXAMPLE",
    "InstanceId": "i-1234567890abcdef0"
```

```
}
```

- 자세한 API 내용은 명령 참조GetConsoleScreenshot의 섹션을 참조하세요. AWS CLI

### get-default-credit-specification

다음 코드 예시에서는 get-default-credit-specification을 사용하는 방법을 보여 줍니다.

AWS CLI

기본 크레딧 옵션을 설명하려면

다음 get-default-credit-specification 예제에서는 T2 인스턴스의 기본 크레딧 옵션을 설명합니다.

```
aws ec2 get-default-credit-specification \
    --instance-family t2
```

출력:

```
{
    "InstanceFamilyCreditSpecification": {
        "InstanceFamily": "t2",
        "CpuCredits": "standard"
    }
}
```

- 자세한 API 내용은 명령 참조GetDefaultCreditSpecification의 섹션을 참조하세요. AWS CLI

### get-ebs-default-kms-key-id

다음 코드 예시에서는 get-ebs-default-kms-key-id을 사용하는 방법을 보여 줍니다.

AWS CLI

EBS 암호화 CMK 기본값을 설명하려면

다음 get-ebs-default-kms-key-id 예제에서는 AWS 계정의 CMK EBS 암호화 기본값을 설명합니다.

```
aws ec2 get-ebs-default-kms-key-id
```

출력은 별칭 CMK 로 AWS 관리되는 CMK EBS 암호화의 기본값을 보여줍니다`alias/aws/ebs`.

```
{
    "KmsKeyId": "alias/aws/ebs"
}
```

다음 출력은 EBS 암호화에 CMK 대한 사용자 지정을 보여줍니다.

```
{
    "KmsKeyId": "arn:aws:kms:us-
west-2:123456789012:key/0ea3fef3-80a7-4778-9d8c-1c0c6EXAMPLE"
}
```

- 자세한 API 내용은 명령 참조GetEbsDefaultKmsKeyId의 섹션을 참조하세요. AWS CLI

### `get-ebs-encryption-by-default`

다음 코드 예시에서는 `get-ebs-encryption-by-default`을 사용하는 방법을 보여 줍니다.

AWS CLI

기본적으로 EBS 암호화가 활성화되어 있는지 여부를 설명하려면

다음 `get-ebs-encryption-by-default` 예제는 현재 리전의 AWS 계정에 대해 기본적으로 EBS 암호화가 활성화되어 있는지 여부를 나타냅니다.

```
aws ec2 get-ebs-encryption-by-default
```

다음 출력은 기본적으로 EBS 암호화가 비활성화되었음을 나타냅니다.

```
{
    "EbsEncryptionByDefault": false
}
```

다음 출력은 기본적으로 EBS 암호화가 활성화되어 있음을 나타냅니다.

```
{
    "EbsEncryptionByDefault": true
}
```

- 자세한 API 내용은 명령 참조GetEbsEncryptionByDefault의 섹션을 참조하세요. AWS CLI

### get-flow-logs-integration-template

다음 코드 예시에서는 get-flow-logs-integration-template을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon Athena와 VPC 흐름 로그의 통합을 자동화하는 CloudFormation 템플릿을 생성하려면

다음 get-flow-logs-integration-template 예제에서는 Amazon Athena 와 VPC 흐름 로그의 통합을 자동화하는 CloudFormation 템플릿을 생성합니다.

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
```

출력:

```
{
    "Result": "https://DOC-EXAMPLE-BUCKET.s3.us-east-2.amazonaws.com/
VPCFlowLogsIntegrationTemplate_fl-1234567890abcdef0_Wed%20Jul
%2021%2000%3A57%3A56%20UTC%202021.yml"
```

```
}
```

CloudFormation 템플릿 사용에 대한 자세한 내용은 AWS CloudFormation 사용 설명서의 AWS CloudFormation 템플릿 작업을 참조하세요.

Amazon Athena 및 흐름 로그 사용에 대한 자세한 내용은 Amazon Virtual Private Cloud 사용 설명서의 Amazon Athena를 사용하여 흐름 로그 쿼리를 참조하세요.

- 자세한 API 내용은 명령 참조GetFlowLogsIntegrationTemplate의 섹션을 참조하세요. AWS CLI

### `get-groups-for-capacity-reservation`

다음 코드 예시에서는 `get-groups-for-capacity-reservation`을 사용하는 방법을 보여 줍니다.

AWS CLI

용량 예약이 있는 리소스 그룹을 나열하려면

다음 `get-groups-for-capacity-reservation` 예제에서는 지정된 용량 예약이 추가된 리소스 그룹을 나열합니다.

```
aws ec2 get-groups-for-capacity-reservation \
    --capacity-reservation-id cr-1234abcd56EXAMPLE
```

출력:

```
{
    "CapacityReservationsGroup": [
        {
            "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/my-
resource-group",
            "OwnerId": "123456789012"
        }
    ]
}
```

자세한 내용은 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 용량 예약 작업을 참조하세요.

- 자세한 API 내용은 명령 참조GetGroupsForCapacityReservation의 섹션을 참조하세요. AWS CLI

### get-host-reservation-purchase-preview

다음 코드 예시에서는 get-host-reservation-purchase-preview을 사용하는 방법을 보여 줍니다.

AWS CLI

전용 호스트 예약에 대한 구매 미리 보기를 가져오려면

이 예제에서는 계정의 지정된 전용 호스트에 대한 지정된 전용 호스트 예약 비용을 미리 봅니다.

명령:

```
aws ec2 get-host-reservation-purchase-preview --offering-id hro-03f707bf363b6b324 --host-id-set h-013abcd2a00cbd123
```

출력:

```
{
  "TotalHourlyPrice": "1.499",
  "Purchase": [
      {
          "HourlyPrice": "1.499",
          "InstanceFamily": "m4",
          "PaymentOption": "NoUpfront",
          "HostIdSet": [
              "h-013abcd2a00cbd123"
          ],
          "UpfrontPrice": "0.000",
          "Duration": 31536000
      }
  ],
  "TotalUpfrontPrice": "0.000"
}
```

- 자세한 API 내용은 명령 참조GetHostReservationPurchasePreview의 섹션을 참조하세요. AWS CLI

### get-image-block-public-access-state

다음 코드 예시에서는 get-image-block-public-access-state을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 리전AMIs에서 에 대한 퍼블릭 액세스 차단 상태를 가져오려면

다음 `get-image-block-public-access-state` 예제에서는 지정된 리전의 계정 수준에서 AMIs에 대한 퍼블릭 액세스 차단 상태를 가져옵니다.

```
aws ec2 get-image-block-public-access-state \
    --region us-east-1
```

출력:

```
{
    "ImageBlockPublicAccessState": "block-new-sharing"
}
```

자세한 내용은 Amazon EC2 사용 설명서의 에 대한 퍼블릭 액세스 차단AMIs을 참조하세요.

- 자세한 API 내용은 명령 참조GetImageBlockPublicAccessState의 섹션을 참조하세요. AWS CLI

### `get-instance-types-from-instance-requirements`

다음 코드 예시에서는 `get-instance-types-from-instance-requirements`을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 속성과 일치하는 인스턴스 유형을 미리 보려면

다음 `get-instance-types-from-instance-requirements` 예제에서는 먼저 `--generate-cli-skeleton` 파라미터를 사용하여 지정할 수 있는 모든 가능한 속성 목록을 생성하고 목록을 JSON 파일에 저장합니다. 그런 다음 JSON 파일을 사용하여 일치하는 인스턴스 유형을 미리 볼 속성을 사용자 지정합니다.

가능한 모든 속성을 생성하고 출력을 JSON 파일에 직접 저장하려면 다음 명령을 사용합니다.

```
aws ec2 get-instance-types-from-instance-requirements \
    --region us-east-1 \
    --generate-cli-skeleton input > attributes.json
```

출력:

```
{
    "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": ""
}
```

JSON 파일을 구성합니다. `ArchitectureTypes`, `VirtualizationTypes`, `VCpuCount` 및 `MemoryMiB`의 값을 입력해야 합니다. 다른 속성을 생략할 수 있습니다. 생략하면 기본값이 사용됩니다. 각 속성 및 기본값에 대한 설명은 get-instance-types-from-instance-requirements <https://docs.aws.amazon.com/cli/latest/reference/ec2/get-instance-types-from-instance-requirements.html>을 참조하세요.

에 지정된 속성이 있는 인스턴스 유형을 미리 봅니다`attributes.json`. `--cli-input-json` 파라미터를 사용하여 JSON 파일의 이름과 경로를 지정합니다. 다음 요청에서 출력은 테이블 형식입니다.

```
aws ec2 get-instance-types-from-instance-requirements \
```

```
    --cli-input-json file://attributes.json \
    --output table
```

attributes.json 파일의 콘텐츠:

```
{

    "ArchitectureTypes": [
        "x86_64"
    ],
    "VirtualizationTypes": [
        "hvm"
    ],
    "InstanceRequirements": {
        "VCpuCount": {
            "Min": 4,
            "Max": 6
        },
        "MemoryMiB": {
            "Min": 2048
        },
        "InstanceGenerations": [
            "current"
        ]
    }
}
```

출력:

```
-----------------------------------------
|GetInstanceTypesFromInstanceRequirements|
+---------------------------------------+
||             InstanceTypes             ||
|+-------------------------------------+|
||             InstanceType              ||
|+-------------------------------------+|
||  c4.xlarge                          ||
||  c5.xlarge                          ||
||  c5a.xlarge                         ||
||  c5ad.xlarge                        ||
||  c5d.xlarge                         ||
||  c5n.xlarge                         ||
||  d2.xlarge                          ||
```

```
...
```

속성 기반 인스턴스 유형 선택에 대한 자세한 내용은 Amazon EC2 사용 설명서의 속성 기반 인스턴스 유형 선택 작동 방식을 참조하세요.

- 자세한 API 내용은 명령 참조GetInstanceTypesFromInstanceRequirements의 섹션을 참조하세요. AWS CLI

### get-instance-uefi-data

다음 코드 예시에서는 get-instance-uefi-data을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스에서 UEFI 데이터를 검색하려면

다음 get-instance-uefi-data 예제는 인스턴스에서 UEFI 데이터를 검색합니다. 출력이 비어 있으면 인스턴스에 UEFI 데이터가 포함되지 않습니다.

```
aws ec2 get-instance-uefi-data \
    --instance-id i-0123456789example
```

출력:

```
{
    "InstanceId": "i-0123456789example",
    "UefiData": "QU1aTlVFRkkf+uLXAAAAAHj5a7fZ9+3dBzxXb/.
    <snipped>
    AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAD4L/J/AODshho="
}
```

자세한 내용은 Amazon EC2 사용 설명서의 UEFI 보안 부팅을 참조하세요.

- 자세한 API 내용은 명령 참조GetInstanceUefiData의 섹션을 참조하세요. AWS CLI

### get-ipam-address-history

다음 코드 예시에서는 get-ipam-address-history을 사용하는 방법을 보여 줍니다.

AWS CLI

의 기록을 가져오려면 CIDR

다음 `get-ipam-address-history` 예제에서는 의 기록을 가져옵니다CIDR.

(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
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon VPC IPAM 사용 설명서의 IP 주소 기록 보기를 참조하세요.

- 자세한 API 내용은 명령 참조GetIpamAddressHistory의 섹션을 참조하세요. AWS CLI

### `get-ipam-discovered-accounts`

다음 코드 예시에서는 `get-ipam-discovered-accounts`을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 검색한 계정을 보려면 IPAM

이 시나리오에서는 가 검색하는 리소스를 소유한 AWS 계정을 보려는 IPAM 위임된 관리자IPAM입니다.

`--discovery-region` 는 모니터링되는 계정 상태를 보려는 IPAM 운영 리전입니다. 예를 들어, IPAM 운영 리전이 세 개 있는 경우 이 요청을 세 번 수행하여 각 특정 리전의 검색에 특정한 타임스탬프를 볼 수 있습니다.

다음 `get-ipam-discovered-accounts` 예제에서는 가 IPAM 검색 중인 리소스를 소유한 AWS 계정을 나열합니다.

```
aws ec2 get-ipam-discovered-accounts \
   --ipam-resource-discovery-id ipam-res-disco-0365d2977fc1672fe \
   --discovery-region us-east-1
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon VPC IPAM 사용 설명서의 조직 외부 계정IPAM과 통합을 참조하세요.

- 자세한 API 내용은 명령 참조GetIpamDiscoveredAccounts의 섹션을 참조하세요. AWS CLI

### `get-ipam-discovered-public-addresses`

다음 코드 예시에서는 `get-ipam-discovered-public-addresses`을 사용하는 방법을 보여 줍니다.

AWS CLI

검색된 퍼블릭 IP 주소를 보려면

이 예제에서는 IPAM 위임된 관리자이며 에서 검색한 리소스의 IP 주소를 보려고 합니다IPAM. 를 사용하여 리소스 검색 ID를 가져올 수 있습니다describe-ipam-resource-discoveries.

다음 `get-ipam-discovered-public-addresses` 예제에서는 리소스 검색에 대해 검색된 퍼블릭 IP 주소를 보여줍니다.

```
aws ec2 get-ipam-discovered-public-addresses \
    --ipam-resource-discovery-id ipam-res-disco-0f4ef577a9f37a162 \
    --address-region us-east-1 \
    --region us-east-1
```

출력:

```
{
    "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"
        }
```

```
}
```

자세한 내용은 Amazon VPC IPAM 사용 설명서의 퍼블릭 IP 인사이트 보기를 참조하세요.

- 자세한 API 내용은 명령 참조GetIpamDiscoveredPublicAddresses의 섹션을 참조하세요. AWS CLI

### get-ipam-discovered-resource-cidrs

다음 코드 예시에서는 get-ipam-discovered-resource-cidrs을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 CIDRs 검색한 IP 주소를 보려면 IPAM

이 예제에서는 가 검색하는 CIDRs 리소스의 IP 주소와 관련된 세부 정보를 보려는 IPAM 위임된 관리자IPAM입니다.

이 요청을 완료하려면:

선택한 리소스 검색은 와 연결되어야 합니다IPAM. --resource-region는 리소스가 생성된 AWS 리전입니다.

다음 get-ipam-discovered-resource-cidrs 예제에서는 IPAM가 검색하는 리소스의 IP 주소를 나열합니다.

```
aws ec2 get-ipam-discovered-resource-cidrs \
   --ipam-resource-discovery-id ipam-res-disco-0365d2977fc1672fe \
   --resource-region us-east-1
```

출력:

```
{
    {
        "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"
        }
    }
}
```

자세한 내용은 Amazon VPC IPAM 사용 설명서의 리소스별 CIDR 사용량 모니터링을 참조하세요.

- 자세한 API 내용은 명령 참조GetIpamDiscoveredResourceCidrs의 섹션을 참조하세요. AWS CLI

### `get-ipam-pool-allocations`

다음 코드 예시에서는 `get-ipam-pool-allocations`을 사용하는 방법을 보여 줍니다.

AWS CLI

IPAM 풀에서 CIDRs 할당을 가져오려면

다음 `get-ipam-pool-allocations` 예제에서는 IPAM 풀에서 CIDRs 할당된 를 가져옵니다.

(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
```

출력:

```
{
    "IpamPoolAllocations": [
        {
            "Cidr": "10.0.0.0/16",
            "IpamPoolAllocationId": "ipam-pool-
alloc-0e6186d73999e47389266a5d6991e6220",
            "ResourceType": "custom",
            "ResourceOwner": "123456789012"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetIpamPoolAllocations의 섹션을 참조하세요. AWS CLI

### get-ipam-pool-cidrs

다음 코드 예시에서는 get-ipam-pool-cidrs을 사용하는 방법을 보여 줍니다.

AWS CLI

IPAM 풀에 CIDRs 프로비저닝하려면

다음 get-ipam-pool-cidrs 예제에서는 IPAM 풀에 CIDRs 프로비저닝된 를 가져옵니다.

(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.*
```

출력:

```
{
    "IpamPoolCidr": {
        "Cidr": "10.0.0.0/24",
        "State": "provisioned"
    }
}
```

- 자세한 API 내용은 명령 참조GetIpamPoolCidrs의 섹션을 참조하세요. AWS CLI

### get-ipam-resource-cidrs

다음 코드 예시에서는 get-ipam-resource-cidrs을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 CIDRs 할당된 을 가져오려면

다음 get-ipam-resource-cidrs 예제에서는 리소스에 CIDRs 할당된 를 가져옵니다.

(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
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon VPC IPAM 사용 설명서의 리소스별 CIDR 사용량 모니터링을 참조하세요.

- 자세한 API 내용은 명령 참조GetIpamResourceCidrs의 섹션을 참조하세요. AWS CLI

### `get-launch-template-data`

다음 코드 예시에서는 `get-launch-template-data`을 사용하는 방법을 보여 줍니다.

AWS CLI

시작 템플릿의 인스턴스 데이터를 가져오려면

이 예제는 지정된 인스턴스에 대한 데이터를 가져오고 --query 옵션을 사용하여 에서 콘텐츠를 반환합니다LaunchTemplateData. 출력을 새로운 시작 템플릿이나 시작 템플릿 버전을 생성하기 위한 기본 템플릿으로 사용할 수 있습니다.

명령:

```
aws ec2 get-launch-template-data --instance-id i-0123d646e8048babc --query
 'LaunchTemplateData'
```

출력:

```
{
      "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetLaunchTemplateData의 섹션을 참조하세요. AWS CLI

### get-managed-prefix-list-associations

다음 코드 예시에서는 get-managed-prefix-list-associations을 사용하는 방법을 보여 줍니다.

AWS CLI

접두사 목록 연결을 가져오려면

다음 get-managed-prefix-list-associations 예제에서는 지정된 접두사 목록과 연결된 리소스를 가져옵니다.

```
aws ec2 get-managed-prefix-list-associations \
    --prefix-list-id pl-0123456abcabcabc1
```

출력:

```
{
    "PrefixListAssociations": [
        {
            "ResourceId": "sg-0abc123456abc12345",
            "ResourceOwner": "123456789012"
        }
    ]
```

```
}
```

자세한 내용은 Amazon VPC 사용 설명서의 관리형 접두사 목록을 참조하세요.

- 자세한 API 내용은 명령 참조GetManagedPrefixListAssociations의 섹션을 참조하세요. AWS CLI

### get-managed-prefix-list-entries

다음 코드 예시에서는 get-managed-prefix-list-entries을 사용하는 방법을 보여 줍니다.

AWS CLI

접두사 목록의 항목을 가져오려면

다음은 지정된 접두사 목록의 항목을 get-managed-prefix-list-entries 가져옵니다.

```
aws ec2 get-managed-prefix-list-entries \
    --prefix-list-id pl-0123456abcabcabc1
```

출력:

```
{
    "Entries": [
        {
            "Cidr": "10.0.0.0/16",
            "Description": "vpc-a"
        },
        {
            "Cidr": "10.2.0.0/16",
            "Description": "vpc-b"
        }
    ]
}
```

자세한 내용은 Amazon VPC 사용 설명서의 관리형 접두사 목록을 참조하세요.

- 자세한 API 내용은 명령 참조GetManagedPrefixListEntries의 섹션을 참조하세요. AWS CLI

### get-network-insights-access-scope-analysis-findings

다음 코드 예시에서는 get-network-insights-access-scope-analysis-findings을 사용하는 방법을 보여 줍니다.

AWS CLI

Network Insights 액세스 범위 분석 결과를 얻으려면

다음 `get-network-insights-access-scope-analysis-findings` 예제에서는 AWS 계정에서 선택한 범위 분석 결과를 가져옵니다.

```
aws ec2 get-network-insights-access-scope-analysis-findings \
    --region us-east-1 \
    --network-insights-access-scope-analysis-id nis \
    --nis-123456789111
```

출력:

```
{
    "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]
            ]
        }
    }
]
}
```

자세한 내용은 Network Access Analyzer 안내서의 를 사용하여 AWS CLI Network Access Analyzer 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조GetNetworkInsightsAccessScopeAnalysisFindings의 섹션을 참조하세요. AWS CLI

### `get-network-insights-access-scope-content`

다음 코드 예시에서는 `get-network-insights-access-scope-content`을 사용하는 방법을 보여 줍니다.

AWS CLI

Network Insights 액세스 범위 콘텐츠를 가져오려면

다음 `get-network-insights-access-scope-content` 예제는 AWS 계정에서 선택한 범위 분석 ID의 콘텐츠를 가져옵니다.

```
aws ec2 get-network-insights-access-scope-content \
    --region us-east-1 \
    --network-insights-access-scope-id nis-123456789222
```

출력:

```
{
    "NetworkInsightsAccessScopeContent": {
        "NetworkInsightsAccessScopeId": "nis-123456789222",
        "MatchPaths": [
            {
                "Source": {
                    "ResourceStatement": {
                        "ResourceTypes": [
                            "AWS::EC2::NetworkInterface"
                        ]
                    }
                },
                "Destination": {
                    "ResourceStatement": {
                        "ResourceTypes": [
                            "AWS::EC2::InternetGateway"
                        ]
                    }
                }
            }
        ]
    }
}
```

자세한 내용은 Network Access Analyzer 가이드의 를 사용하여 AWS CLI Network Access Analyzer 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조GetNetworkInsightsAccessScopeContent의 섹션을 참조하세요. AWS CLI

### `get-password-data`

다음 코드 예시에서는 `get-password-data`을 사용하는 방법을 보여 줍니다.

AWS CLI

암호화된 암호를 가져오려면

이 예제에서는 암호화된 암호를 가져옵니다.

명령:

```
aws ec2 get-password-data --instance-id i-1234567890abcdef0
```

출력:

```
{
    "InstanceId": "i-1234567890abcdef0",
    "Timestamp": "2013-08-07T22:18:38.000Z",
    "PasswordData": "gSlJFq+VpcZXqy+iktxMF6NyxQ4qCrT4+gaOuNOenX1MmgXPTj7XEXAMPLE
UQ+YeFfb+L1U4C4AKv652Ux1iRB3CPTYP7WmU3TUnhsuBd+p6LVk7T2lKUml6OXbk6WPW1VYYm/TRPB1
e1DQ7PY4an/DgZT4mwcpRFigzhniQgDDeO1InvSDcwoUTwNs0Y1S8ouri2W4n5GNlriM3Q0AnNVelVz/
53TkDtxbNoU606M1gK9zUWSxqEgwvbV2j8c5rP0WCuaMWSFl4ziDu4bd7q+4RSyi8NUsVWnKZ4aEZffu
DPGzKrF5yLlf3etP2L4ZR6CvG7K1hx7VKOQVN32Dajw=="
}
```

복호화된 암호를 가져오려면

이 예제에서는 암호 해독된 암호를 가져옵니다.

명령:

```
aws ec2 get-password-data --instance-id  i-1234567890abcdef0 --priv-launch-key C:
\Keys\MyKeyPair.pem
```

출력:

```
{
    "InstanceId": "i-1234567890abcdef0",
    "Timestamp": "2013-08-30T23:18:05.000Z",
    "PasswordData": "&ViJ652e*u"
}
```

- 자세한 API 내용은 명령 참조GetPasswordData의 섹션을 참조하세요. AWS CLI

### `get-reserved-instances-exchange-quote`

다음 코드 예시에서는 `get-reserved-instances-exchange-quote`을 사용하는 방법을 보여 줍니다.

AWS CLI

컨버터블 예약 인스턴스 교환에 대한 견적을 가져오려면

이 예제에서는 지정된 컨버터블 예약 인스턴스에 대한 교환 정보를 가져옵니다.

명령:

```
aws ec2 get-reserved-instances-exchange-quote --reserved-
instance-ids 7b8750c3-397e-4da4-bbcb-a45ebexample --target-
configurations OfferingId=6fea5434-b379-434c-b07b-a7abexample
```

출력:

```
{
  "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"
  }
}
```

- 자세한 API 내용은 명령 참조GetReservedInstancesExchangeQuote의 섹션을 참조하세요. AWS CLI

### get-security-groups-for-vpc

다음 코드 예시에서는 get-security-groups-for-vpc을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 에서 네트워크 인터페이스와 연결할 수 있는 보안 그룹을 봅니다VPC.

다음 get-security-groups-for-vpc 예제는 의 네트워크 인터페이스와 연결할 수 있는 보안 그룹을 보여줍니다VPC.

```
aws ec2 get-security-groups-for-vpc \
    --vpc-id vpc-6c31a611 \
    --region us-east-1
```

출력:

```
{
    "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"
        }
}
```

- 자세한 API 내용은 명령 참조GetSecurityGroupsForVpc의 섹션을 참조하세요. AWS CLI

### get-serial-console-access-status

다음 코드 예시에서는 get-serial-console-access-status을 사용하는 방법을 보여 줍니다.

AWS CLI

직렬 콘솔에 대한 계정 액세스 상태를 보려면

다음 get-serial-console-access-status 예제에서는 계정에 직렬 콘솔 액세스가 활성화되어 있는지 여부를 결정합니다.

```
aws ec2 get-serial-console-access-status
```

출력:

```
{
    "SerialConsoleAccessEnabled": true
}
```

자세한 내용은 Amazon EC2 사용 설명서의 EC2 직렬 콘솔을 참조하세요.

- 자세한 API 내용은 명령 참조GetSerialConsoleAccessStatus의 섹션을 참조하세요. AWS CLI

### get-snapshot-block-public-access-state

다음 코드 예시에서는 get-snapshot-block-public-access-state을 사용하는 방법을 보여 줍니다.

AWS CLI

스냅샷에 대한 퍼블릭 액세스 차단의 현재 상태를 가져오려면

다음 get-snapshot-block-public-access-state 예제에서는 스냅샷에 대한 퍼블릭 액세스 차단의 현재 상태를 가져옵니다.

```
aws ec2 get-snapshot-block-public-access-state
```

출력:

```
{
    "State": "block-all-sharing"
}
```

자세한 내용은 Amazon EBS 사용 설명서의 스냅샷에 대한 퍼블릭 액세스 차단을 참조하세요.

- 자세한 API 내용은 명령 참조GetSnapshotBlockPublicAccessState의 섹션을 참조하세요. AWS CLI

### get-spot-placement-scores

다음 코드 예시에서는 get-spot-placement-scores을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 요구 사항에 대한 스팟 배치 점수를 계산하려면

다음 get-spot-placement-scores 예제에서는 먼저 파라미터를 사용하여 스팟 배치 점수 구성에 지정할 수 있는 가능한 모든 --generate-cli-skeleton 파라미터 목록을 생성하고 목록을 JSON 파일에 저장합니다. 그런 다음 JSON 파일은 스팟 배치 점수를 계산하는 데 사용할 요구 사항을 구성하는 데 사용됩니다.

스팟 배치 점수 구성에 지정할 수 있는 가능한 모든 파라미터를 생성하고 출력을 JSON 파일에 직접 저장합니다.

```
aws ec2 get-spot-placement-scores \
    --region us-east-1 \
    --generate-cli-skeleton input > attributes.json
```

출력:

```
{
    "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": ""
}
```

JSON 파일을 구성합니다. `TargetCapacity`의 값을 제공해야 합니다. 각 파라미터와 기본값에 대한 설명은 스팟 배치 점수 계산(AWS CLI) <https://docs.aws.amazon.com/AWS EC2/latest/UserGuide/spot-placement-score.html#calculate-sps-cli>을 참조하세요.

에 지정된 요구 사항에 대한 스팟 배치 점수를 계산합니다`attributes.json`. `--cli-input-json` 파라미터를 사용하여 JSON 파일의 이름과 경로를 지정합니다.

```
aws ec2 get-spot-placement-scores \
    --region us-east-1 \
```

```
    --cli-input-json file://attributes.json
```

SingleAvailabilityZone 이 로 설정false되거나 생략된 경우의 출력(생략된 경우 기본값은 false). 점수가 매겨진 리전 목록이 반환됩니다.

```
"Recommendation": [
    {
        "Region": "us-east-1",
        "Score": 7
    },
    {
        "Region": "us-west-1",
        "Score": 5
    },
   ...
```

SingleAvailabilityZone 가 로 설정된 경우 출력true. 점수가 매겨진 SingleAvailability 영역 목록이 반환됩니다.

```
"Recommendation": [
    {
        "Region": "us-east-1",
        "AvailabilityZoneId": "use1-az1"
        "Score": 8
    },
    {
        "Region": "us-east-1",
        "AvailabilityZoneId": "usw2-az3"
        "Score": 6
    },
   ...
```

스팟 배치 점수 계산에 대한 자세한 내용 및 예를 들어 구성은 Amazon EC2 사용 설명서의 스팟 배치 점수 계산을 참조하세요.

- 자세한 API 내용은 명령 참조GetSpotPlacementScores의 섹션을 참조하세요. AWS CLI

### get-subnet-cidr-reservations

다음 코드 예시에서는 get-subnet-cidr-reservations을 사용하는 방법을 보여 줍니다.

AWS CLI

서브넷 CIDR 예약에 대한 정보를 가져오려면

다음 `get-subnet-cidr-reservations` 예제에서는 지정된 서브넷 CIDR 예약에 대한 정보를 표시합니다.

```
aws ec2 get-subnet-cidr-reservations \
    --subnet-id subnet-03c51e2e6cEXAMPLE
```

출력:

```
{
    "SubnetIpv4CidrReservations": [
        {
            "SubnetCidrReservationId": "scr-044f977c4eEXAMPLE",
            "SubnetId": "subnet-03c51e2e6cEXAMPLE",
            "Cidr": "10.1.0.16/28",
            "ReservationType": "prefix",
            "OwnerId": "123456789012"
        }
    ],
    "SubnetIpv6CidrReservations": []
}
```

자세한 내용은 Amazon VPC 사용 설명서의 서브넷 CIDR 예약을 참조하세요.

- 자세한 API 내용은 명령 참조GetSubnetCidrReservations의 섹션을 참조하세요. AWS CLI

### `get-transit-gateway-attachment-propagations`

다음 코드 예시에서는 `get-transit-gateway-attachment-propagations`을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 리소스 연결에서 라우팅을 전파하는 라우팅 테이블을 나열하려면

다음 `get-transit-gateway-attachment-propagations` 예제에서는 지정된 리소스 연결에서 라우팅을 전파하는 라우팅 테이블을 나열합니다.

```
aws ec2 get-transit-gateway-attachment-propagations \
```

```
    --transit-gateway-attachment-id tgw-attach-09fbd47ddfEXAMPLE
```

출력:

```
{
    "TransitGatewayAttachmentPropagations": [
        {
            "TransitGatewayRouteTableId": "tgw-rtb-0882c61b97EXAMPLE",
            "State": "enabled"
        }
    ]
}
```

자세한 내용은 Transit Gateways Guide의 Transit Gateway route table을 참조하세요.

- 자세한 API 내용은 명령 참조GetTransitGatewayAttachmentPropagations의 섹션을 참조하세요. AWS CLI

### get-transit-gateway-multicast-domain-associations

다음 코드 예시에서는 get-transit-gateway-multicast-domain-associations을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 멀티캐스트 도메인 연결에 대한 정보를 보려면

다음 get-transit-gateway-multicast-domain-associations 예제에서는 지정된 멀티캐스트 도메인에 대한 연결을 반환합니다.

```
aws ec2 get-transit-gateway-multicast-domain-associations \
    --transit-gateway-multicast-domain-id tgw-mcast-domain-0c4905cef7EXAMPLE
```

출력:

```
{
    "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"
            }
        }
    ]
}
```

자세한 내용은 Transit Gateways 가이드의 멀티캐스트 도메인 관리를 참조하세요.

- 자세한 API 내용은 명령 참조GetTransitGatewayMulticastDomainAssociations의 섹션을 참조하세요. AWS CLI

### get-transit-gateway-prefix-list-references

다음 코드 예시에서는 get-transit-gateway-prefix-list-references을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 라우팅 테이블에서 접두사 목록 참조를 가져오려면

다음 get-transit-gateway-prefix-list-references 예제에서는 지정된 전송 게이트웨이 라우팅 테이블에 대한 접두사 목록 참조를 가져오고 특정 접두사 목록의 ID로 필터링합니다.

```
aws ec2 get-transit-gateway-prefix-list-references \
    --transit-gateway-route-table-id tgw-rtb-0123456789abcd123 \
    --filters Name=prefix-list-id,Values=pl-11111122222222333
```

출력:

```
{
    "TransitGatewayPrefixListReferences": [
        {
            "TransitGatewayRouteTableId": "tgw-rtb-0123456789abcd123",
            "PrefixListId": "pl-11111122222222333",
            "PrefixListOwnerId": "123456789012",
            "State": "available",
            "Blackhole": false,
            "TransitGatewayAttachment": {
                "TransitGatewayAttachmentId": "tgw-attach-aabbccddaabbccaab",
                "ResourceType": "vpc",
                "ResourceId": "vpc-112233445566aabbc"
            }
        }
    ]
}
```

자세한 내용은 Transit Gateways 가이드의 접두사 목록 참조를 참조하세요.

- 자세한 API 내용은 명령 참조GetTransitGatewayPrefixListReferences의 섹션을 참조하세요. AWS CLI

### get-transit-gateway-route-table-associations

다음 코드 예시에서는 get-transit-gateway-route-table-associations을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 전송 게이트웨이 라우팅 테이블의 연결에 대한 정보를 가져오려면

다음 get-transit-gateway-route-table-associations 예제에서는 지정된 전송 게이트웨이 라우팅 테이블의 연결에 대한 정보를 표시합니다.

```
aws ec2 get-transit-gateway-route-table-associations \
    --transit-gateway-route-table-id tgw-rtb-0a823edbdeEXAMPLE
```

출력:

```
{
    "Associations": [
        {
            "TransitGatewayAttachmentId": "tgw-attach-09b52ccdb5EXAMPLE",
            "ResourceId": "vpc-4d7de228",
            "ResourceType": "vpc",
            "State": "associating"
        }
    ]
}
```

자세한 내용은 Transit Gateways Guide의 Transit Gateway route table을 참조하세요.

- 자세한 API 내용은 명령 참조GetTransitGatewayRouteTableAssociations의 섹션을 참조하세요. AWS CLI

### get-transit-gateway-route-table-propagations

다음 코드 예시에서는 get-transit-gateway-route-table-propagations을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 전송 게이트웨이 라우팅 테이블의 라우팅 테이블 전파에 대한 정보를 표시하려면

다음 `get-transit-gateway-route-table-propagations` 예제에서는 지정된 라우팅 테이블에 대한 라우팅 테이블 전파를 반환합니다.

```
aws ec2 get-transit-gateway-route-table-propagations \
    --transit-gateway-route-table-id tgw-rtb-002573ed1eEXAMPLE
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Transit Gateways Guide의 Transit Gateway route table을 참조하세요.

- 자세한 API 내용은 명령 참조GetTransitGatewayRouteTablePropagations의 섹션을 참조하세요. AWS CLI

### get-verified-access-endpoint-policy

다음 코드 예시에서는 get-verified-access-endpoint-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

엔드포인트의 Verified Access 정책을 가져오려면

다음 get-verified-access-endpoint-policy 예제에서는 지정된 엔드포인트의 Verified Access 정책을 가져옵니다.

```
aws ec2 get-verified-access-endpoint-policy \
    --verified-access-endpoint-id vae-066fac616d4d546f2
```

출력:

```
{
    "PolicyEnabled": true,
    "PolicyDocument": "permit(principal,action,resource)\nwhen
 {\n    context.identity.groups.contains(\"finance\") &&\n
 context.identity.email_verified == true\n};"
}
```

자세한 내용은 Verified Access 사용 설명서의 Verified Access 정책을 참조하세요. AWS

- 자세한 API 내용은 명령 참조GetVerifiedAccessEndpointPolicy의 섹션을 참조하세요. AWS CLI

### get-verified-access-group-policy

다음 코드 예시에서는 get-verified-access-group-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

그룹의 Verified Access 정책을 가져오려면

다음 get-verified-access-group-policy 예제에서는 지정된 그룹의 Verified Access 정책을 가져옵니다.

```
aws ec2 get-verified-access-group-policy \
```

```
    --verified-access-group-id vagr-0dbe967baf14b7235
```

출력:

```
{
    "PolicyEnabled": true,
    "PolicyDocument": "permit(principal,action,resource)\nwhen
 {\n    context.identity.groups.contains(\"finance\") &&\n
 context.identity.email_verified == true\n};"
}
```

자세한 내용은 Verified Access 사용 설명서의 Verified Access 그룹을 참조하세요. AWS

- 자세한 API 내용은 명령 참조GetVerifiedAccessGroupPolicy의 섹션을 참조하세요. AWS CLI

### get-vpn-connection-device-sample-configuration

다음 코드 예시에서는 get-vpn-connection-device-sample-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

샘플 구성 파일을 다운로드하려면

다음 get-vpn-connection-device-sample-configuration 예제에서는 지정된 샘플 구성 파일을 다운로드합니다. 샘플 구성 파일이 있는 게이트웨이 디바이스를 나열하려면 get-vpn-connection-device-types 명령을 호출합니다.

```
aws ec2 get-vpn-connection-device-sample-configuration \
    --vpn-connection-id vpn-123456789abc01234 \
    --vpn-connection-device-type-id 5fb390ba
```

출력:

```
{
    "VpnConnectionDeviceSampleConfiguration": "contents-of-the-sample-configuration-
file"
}
```

자세한 내용은 AWS Site-to-Site VPN 사용 설명서의 구성 파일 다운로드를 참조하세요.

- 자세한 API 내용은 명령 참조GetVpnConnectionDeviceSampleConfiguration의 섹션을 참조하세요. AWS CLI

### get-vpn-connection-device-types

다음 코드 예시에서는 get-vpn-connection-device-types을 사용하는 방법을 보여 줍니다.

AWS CLI

샘플 구성 파일이 있는 게이트웨이 디바이스를 나열하려면

다음 get-vpn-connection-device-types 예제에서는 샘플 구성 파일이 있는 Palo Alto Networks의 게이트웨이 디바이스를 나열합니다.

```
aws ec2 get-vpn-connection-device-types \
    --query "VpnConnectionDeviceTypes[?Vendor==`Palo Alto Networks`]"
```

출력:

```
[
    {
        "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+"
    }
]
```

자세한 내용은 AWS Site-to-Site VPN 사용 설명서의 구성 파일 다운로드를 참조하세요.

- 자세한 API 내용은 명령 참조GetVpnConnectionDeviceTypes의 섹션을 참조하세요. AWS CLI

### import-client-vpn-client-certificate-revocation-list

다음 코드 예시에서는 import-client-vpn-client-certificate-revocation-list을 사용하는 방법을 보여 줍니다.

AWS CLI

클라이언트 인증서 취소 목록을 가져오려면

다음 import-client-vpn-client-certificate-revocation-list 예제에서는 로컬 컴퓨터에서 파일의 위치를 지정하여 클라이언트 인증서 취소 목록을 클라이언트 VPN 엔드포인트로 가져옵니다.

```
aws ec2 import-client-vpn-client-certificate-revocation-list \
    --certificate-revocation-list file:///path/to/crl.pem \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde
```

출력:

```
{
    "Return": true
}
```

자세한 내용은 클라이언트 관리자 안내서의 클라이언트 인증서 취소 목록을 참조하세요. AWS VPN

- 자세한 API 내용은 명령 참조ImportClientVpnClientCertificateRevocationList의 섹션을 참조하세요. AWS CLI

### import-image

다음 코드 예시에서는 import-image을 사용하는 방법을 보여 줍니다.

AWS CLI

로 VM 이미지 파일을 가져오려면 AMI

다음 import-image 예제에서는 지정된 를 가져옵니다OVA.

```
aws ec2 import-image \
  --disk-containers Format=ova,UserBucket="{S3Bucket=my-import-bucket,S3Key=vms/my-server-vm.ova}"
```

출력:

```
{
    "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"
}
```

- 자세한 API 내용은 명령 참조ImportImage의 섹션을 참조하세요. AWS CLI

### import-key-pair

다음 코드 예시에서는 import-key-pair을 사용하는 방법을 보여 줍니다.

AWS CLI

퍼블릭 키를 가져오려면

먼저 원하는 도구로 키 페어를 생성합니다. 예를 들어 이 ssh-keygen 명령을 사용합니다.

명령:

```
ssh-keygen -t rsa -C "my-key" -f ~/.ssh/my-key
```

출력:

```
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.
...
```

이 예제 명령은 지정된 퍼블릭 키를 가져옵니다.

명령:

```
aws ec2 import-key-pair --key-name "my-key" --public-key-material fileb://~/.ssh/my-key.pub
```

출력:

```
{
  "KeyName": "my-key",
  "KeyFingerprint": "1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca"
}
```

- 자세한 API 내용은 명령 참조ImportKeyPair의 섹션을 참조하세요. AWS CLI

### import-snapshot

다음 코드 예시에서는 import-snapshot을 사용하는 방법을 보여 줍니다.

AWS CLI

스냅샷을 가져오려면

다음 import-snapshot 예제에서는 지정된 디스크를 스냅샷으로 가져옵니다.

```
aws ec2 import-snapshot \
    --description "My server VMDK" \
    --disk-container Format=VMDK,UserBucket={S3Bucket=my-import-bucket,S3Key=vms/my-server-vm.vmdk}
```

출력:

```
{
    "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"
        }
    }
}
```

- 자세한 API 내용은 명령 참조ImportSnapshot의 섹션을 참조하세요. AWS CLI

### list-images-in-recycle-bin

다음 코드 예시에서는 list-images-in-recycle-bin을 사용하는 방법을 보여 줍니다.

AWS CLI

휴지통에 이미지를 나열하려면

다음 list-images-in-recycle-bin 예제에서는 현재 휴지통에 보관된 모든 이미지를 나열합니다.

```
aws ec2 list-images-in-recycle-bin
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Elastic Compute Cloud 사용 설명서의 휴지통AMIs에서 복구를 참조하세요.

- 자세한 API 내용은 명령 참조ListImagesInRecycleBin의 섹션을 참조하세요. AWS CLI

### `list-snapshots-in-recycle-bin`

다음 코드 예시에서는 `list-snapshots-in-recycle-bin`을 사용하는 방법을 보여 줍니다.

AWS CLI

휴지통에서 스냅샷을 보려면

다음 `list-snapshots-in-recycle-bin` 예제에서는 스냅샷 ID, 스냅샷 설명, 스냅샷이 생성된 볼륨의 ID, 스냅샷이 삭제되고 휴지통에 입력된 날짜 및 시간, 보존 기간이 만료된 날짜 및 시간을 포함하여 휴지통의 스냅샷에 대한 정보를 나열합니다.

```
aws ec2 list-snapshots-in-recycle-bin \
    --snapshot-id snap-01234567890abcdef
```

출력:

```
{
    "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"
        }
    ]
}
```

Amazon용 휴지통에 대한 자세한 내용은 Amazon EC2 사용 설명서의 휴지통에서 스냅샷 복구를 EBS참조하세요.

- 자세한 API 내용은 명령 참조ListSnapshotsInRecycleBin의 섹션을 참조하세요. AWS CLI

### `lock-snapshot`

다음 코드 예시에서는 `lock-snapshot`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 거버넌스 모드에서 스냅샷을 잠그는 방법

다음 `lock-snapshot` 예제에서는 거버넌스 모드에서 지정된 스냅샷을 잠급니다.

```
aws ec2 lock-snapshot \
    --snapshot-id snap-0b5e733b4a8df6e0d \
    --lock-mode governance \
    --lock-duration 365
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon EBS 사용 설명서의 스냅샷 잠금을 참조하세요.

예제 2: 규정 준수 모드에서 스냅샷을 잠그려면

다음 `lock-snapshot` 예제에서는 지정된 스냅샷을 규정 준수 모드에서 잠급니다.

```
aws ec2 lock-snapshot \
    --snapshot-id snap-0163a8524c5b9901f \
    --lock-mode compliance \
    --cool-off-period 24 \
    --lock-duration 365
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon EBS 사용 설명서의 스냅샷 잠금을 참조하세요.

- 자세한 API 내용은 명령 참조LockSnapshot의 섹션을 참조하세요. AWS CLI

### `modify-address-attribute`

다음 코드 예시에서는 `modify-address-attribute`을 사용하는 방법을 보여 줍니다.

AWS CLI

탄력적 IP 주소와 연결된 도메인 이름 속성을 수정하려면

다음 `modify-address-attribute` 예제에서는 탄력적 IP 주소의 도메인 이름 속성을 수정합니다.

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
```

출력:

```
{
```

```
    "Addresses": [
        {
            "PublicIp": "192.0.2.0",
            "AllocationId": "eipalloc-abcdef01234567890",
            "PtrRecord": "example.net."
            "PtrRecordUpdate": {
                "Value": "example.com.",
                "Status": "PENDING"
        }
    ]
}
```

보류 중인 변경 사항을 모니터링하고 탄력적 IP 주소의 수정된 속성을 보려면 명령 참조describe-addresses-attribute의 섹션을 참조하세요. AWS CLI

- 자세한 API 내용은 명령 참조ModifyAddressAttribute의 섹션을 참조하세요. AWS CLI

### `modify-availability-zone-group`

다음 코드 예시에서는 `modify-availability-zone-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

영역 그룹을 활성화하려면

다음 `modify-availability-zone-group` 예제에서는 지정된 영역 그룹을 활성화합니다.

```
aws ec2 modify-availability-zone-group \
    --group-name us-west-2-lax-1 \
    --opt-in-status opted-in
```

출력:

```
{
    "Return": true
}
```

자세한 내용은 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 리전 및 영역을 참조하세요.

- 자세한 API 내용은 명령 참조ModifyAvailabilityZoneGroup의 섹션을 참조하세요. AWS CLI

### modify-capacity-reservation-fleet

다음 코드 예시에서는 modify-capacity-reservation-fleet을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 용량 예약 플릿의 총 목표 용량을 수정하려면

다음 modify-capacity-reservation-fleet 예제에서는 지정된 용량 예약 플릿의 총 목표 용량을 수정합니다. 용량 예약 플릿의 총 목표 용량을 수정하면 플릿이 자동으로 새 용량 예약을 생성하거나, 새 총 목표 용량을 충족하도록 플릿의 기존 용량 예약을 수정 또는 취소합니다. modifying 상태인 동안에는 플릿에 대해 추가 수정을 시도할 수 없습니다.

```
aws ec2 modify-capacity-reservation-fleet \
    --capacity-reservation-fleet-id crf-01234567890abcedf \
    --total-target-capacity 160
```

출력:

```
{
    "Return": true
}
```

예제 2: 용량 예약 플릿의 종료 날짜를 수정하려면

다음 modify-capacity-reservation-fleet 예제에서는 지정된 용량 예약 플릿의 종료 날짜를 수정합니다. 플릿의 종료 날짜를 수정하면 모든 개별 용량 예약의 종료 날짜가 그에 따라 업데이트됩니다. modifying 상태인 동안에는 플릿에 대해 추가 수정을 시도할 수 없습니다.

```
aws ec2 modify-capacity-reservation-fleet \
    --capacity-reservation-fleet-id crf-01234567890abcedf \
    --end-date 2022-07-04T23:59:59.000Z
```

출력:

```
{
    "Return": true
}
```

용량 예약 플릿에 대한 자세한 내용은 Amazon EC2 사용 설명서의 용량 예약 플릿을 참조하세요.

- 자세한 API 내용은 명령 참조ModifyCapacityReservationFleet의 섹션을 참조하세요. AWS CLI

### modify-capacity-reservation

다음 코드 예시에서는 modify-capacity-reservation을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 기존 용량 예약에서 예약한 인스턴스 수 변경

다음 modify-capacity-reservation 예제에서는 용량 예약이 용량을 예약하는 인스턴스 수를 변경합니다.

```
aws ec2 modify-capacity-reservation \
    --capacity-reservation-id cr-1234abcd56EXAMPLE \
    --instance-count 5
```

출력:

```
{
    "Return": true
}
```

예제 2: 기존 용량 예약의 종료 날짜 및 시간 변경

다음 modify-capacity-reservation 예제에서는 지정된 날짜 및 시간에 종료되도록 기존 용량 예약을 수정합니다.

```
aws ec2 modify-capacity-reservation \
    --capacity-reservation-id cr-1234abcd56EXAMPLE \
    --end-date-type limited \
    --end-date 2019-08-31T23:59:59Z
```

자세한 내용은 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 용량 예약 수정을 참조하세요.

- 자세한 API 내용은 명령 참조ModifyCapacityReservation의 섹션을 참조하세요. AWS CLI

### modify-client-vpn-endpoint

다음 코드 예시에서는 modify-client-vpn-endpoint을 사용하는 방법을 보여 줍니다.

AWS CLI

클라이언트 VPN 엔드포인트를 수정하려면

다음 modify-client-vpn-endpoint 예제에서는 지정된 클라이언트 VPN 엔드포인트에 대한 클라이언트 연결 로깅을 활성화합니다.

```
aws ec2 modify-client-vpn-endpoint \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \
    --connection-log-options Enabled=true,CloudwatchLogGroup=ClientVPNLogs
```

출력:

```
{
    "Return": true
}
```

자세한 내용은 클라이언트 관리자 안내서의 클라이언트 VPN 엔드포인트를 참조하세요. AWS VPN

- 자세한 API 내용은 명령 참조ModifyClientVpnEndpoint의 섹션을 참조하세요. AWS CLI

### modify-default-credit-specification

다음 코드 예시에서는 modify-default-credit-specification을 사용하는 방법을 보여 줍니다.

AWS CLI

기본 크레딧 옵션을 수정하려면

다음 modify-default-credit-specification 예제에서는 T2 인스턴스의 기본 크레딧 옵션을 수정합니다.

```
aws ec2 modify-default-credit-specification \
    --instance-family t2 \
    --cpu-credits unlimited
```

출력:

```
{
    "InstanceFamilyCreditSpecification": {
```

```
        "InstanceFamily": "t2",
        "CpuCredits": "unlimited"
    }
}
```

- 자세한 API 내용은 명령 참조ModifyDefaultCreditSpecification의 섹션을 참조하세요. AWS CLI

### modify-ebs-default-kms-key-id

다음 코드 예시에서는 modify-ebs-default-kms-key-id을 사용하는 방법을 보여 줍니다.

AWS CLI

EBS 암호화 CMK 기본값을 설정하려면

다음 modify-ebs-default-kms-key-id 예제에서는 지정된 를 현재 리전의 AWS 계정에 대한 CMK EBS 암호화 기본값CMK으로 설정합니다.

```
aws ec2 modify-ebs-default-kms-key-id \
    --kms-key-id alias/my-cmk
```

출력:

```
{
    "KmsKeyId": "arn:aws:kms:us-
west-2:123456789012:key/0ea3fef3-80a7-4778-9d8c-1c0c6EXAMPLE"
}
```

- 자세한 API 내용은 명령 참조ModifyEbsDefaultKmsKeyId의 섹션을 참조하세요. AWS CLI

### modify-fleet

다음 코드 예시에서는 modify-fleet을 사용하는 방법을 보여 줍니다.

AWS CLI

EC2플릿 크기 조정

다음 modify-fleet 예제에서는 지정된 EC2플릿의 대상 용량을 수정합니다. 지정된 값이 현재 용량보다 크면 EC2 플릿이 추가 인스턴스를 시작합니다. 지정된 값이 현재 용량보다 작으면 EC2 플

릿은 모든 미결 요청을 취소하고 종료 정책이 이면 terminateEC2플릿은 새 목표 용량을 초과하는 인스턴스를 종료합니다.

```
aws ec2 modify-fleet \
    --fleet-ids fleet-12a34b55-67cd-8ef9-ba9b-9208dEXAMPLE \
    --target-capacity-specification TotalTargetCapacity=5
```

출력:

```
{
    "Return": true
}
```

자세한 내용은 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 EC2 플릿 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ModifyFleet의 섹션을 참조하세요. AWS CLI

### modify-fpga-image-attribute

다음 코드 예시에서는 modify-fpga-image-attribute을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon FPGA 이미지의 속성을 수정하려면

이 예제에서는 지정된 에 대한 계정 ID에 123456789012 대한 로드 권한을 추가합니다AFI.

명령:

```
aws ec2 modify-fpga-image-attribute --attribute loadPermission --fpga-image-
id afi-0d123e123bfc85abc --load-permission Add=[{UserId=123456789012}]
```

출력:

```
{
  "FpgaImageAttribute": {
      "FpgaImageId": "afi-0d123e123bfc85abc",
      "LoadPermissions": [
          {
              "UserId": "123456789012"
```

```
            }
        ]
    }
}
```

- 자세한 API 내용은 명령 참조ModifyFpgaImageAttribute의 섹션을 참조하세요. AWS CLI

### `modify-hosts`

다음 코드 예시에서는 `modify-hosts`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 전용 호스트에 대한 자동 배치를 활성화하려면

다음 `modify-hosts` 예제에서는 전용 호스트에 대한 자동 배치를 활성화하여 인스턴스 유형 구성과 일치하는 모든 대상 미지정 인스턴스 시작을 허용합니다.

```
aws ec2 modify-hosts \
    --host-id h-06c2f189b4EXAMPLE \
    --auto-placement on
```

출력:

```
{
    "Successful": [
        "h-06c2f189b4EXAMPLE"
    ],
    "Unsuccessful": []
}
```

예제 2: 전용 호스트에 대한 호스트 복구를 활성화하려면

다음 `modify-hosts` 예제에서는 지정된 전용 호스트에 대한 호스트 복구를 활성화합니다.

```
aws ec2 modify-hosts \
    --host-id h-06c2f189b4EXAMPLE \
    --host-recovery on
```

출력:

```
{
    "Successful": [
        "h-06c2f189b4EXAMPLE"
    ],
    "Unsuccessful": []
}
```

자세한 내용은 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 전용 호스트 자동 배치 수정을 참조하세요.

- 자세한 API 내용은 명령 참조ModifyHosts의 섹션을 참조하세요. AWS CLI

### modify-id-format

다음 코드 예시에서는 modify-id-format을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 더 긴 ID 형식을 활성화하려면

다음 modify-id-format 예제에서는 instance 리소스 유형에 더 긴 ID 형식을 활성화합니다.

```
aws ec2 modify-id-format \
    --resource instance \
    --use-long-ids
```

리소스의 더 긴 ID 형식을 비활성화하려면

다음 modify-id-format 예제에서는 instance 리소스 유형에 대한 긴 ID 형식을 비활성화합니다.

```
aws ec2 modify-id-format \
    --resource instance \
    --no-use-long-ids
```

다음 modify-id-format 예제에서는 옵트인 기간 내에 있는 지원되는 모든 리소스 유형에 대해 더 긴 ID 형식을 활성화합니다.

```
aws ec2 modify-id-format \
    --resource all-current \
    --use-long-ids
```

- 자세한 API 내용은 명령 참조ModifyIdFormat의 섹션을 참조하세요. AWS CLI

### modify-identity-id-format

다음 코드 예시에서는 modify-identity-id-format을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 역할이 리소스IDs에 더 오래 사용할 수 있도록 활성화하려면

다음 modify-identity-id-format 예제에서는 AWS 계정의 IAM 역할EC2Role이 instance 리소스 유형에 긴 ID 형식을 사용할 수 있습니다.

```
aws ec2 modify-identity-id-format \
    --principal-arn arn:aws:iam::123456789012:role/EC2Role \
    --resource instance \
    --use-long-ids
```

IAM 사용자가 리소스에 IDs 더 오래 사용할 수 있도록 하려면

다음 modify-identity-id-format 예제에서는 AWS 계정의 IAM 사용자가 volume 리소스 유형에 더 긴 ID 형식을 사용할 수 AdminUser 있습니다.

```
aws ec2 modify-identity-id-format \
    --principal-arn arn:aws:iam::123456789012:user/AdminUser \
    --resource volume \
    --use-long-ids
```

다음 modify-identity-id-format 예제AdminUser에서는 AWS 계정의 IAM 사용자가 옵트인 기간 내에 지원되는 모든 리소스 유형에 더 긴 ID 형식을 사용할 수 있습니다.

```
aws ec2 modify-identity-id-format \
    --principal-arn arn:aws:iam::123456789012:user/AdminUser \
    --resource all-current \
    --use-long-ids
```

- 자세한 API 내용은 명령 참조ModifyIdentityIdFormat의 섹션을 참조하세요. AWS CLI

### modify-image-attribute

다음 코드 예시에서는 modify-image-attribute을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: AMI 공개

다음 modify-instance-attribute 예제에서는 지정된 를 AMI 퍼블릭으로 만듭니다.

```
aws ec2 modify-image-attribute \
    --image-id ami-5731123e \
    --launch-permission "Add=[{Group=all}]"
```

이 명령은 출력을 생성하지 않습니다.

예제 2: AMI 프라이빗을 만들려면

다음 modify-instance-attribute 예제에서는 지정된 를 AMI 비공개로 설정합니다.

```
aws ec2 modify-image-attribute \
    --image-id ami-5731123e \
    --launch-permission "Remove=[{Group=all}]"
```

이 명령은 출력을 생성하지 않습니다.

예제 3: AWS 계정에 시작 권한을 부여하려면

다음 modify-instance-attribute 예제에서는 지정된 AWS 계정에 시작 권한을 부여합니다.

```
aws ec2 modify-image-attribute \
    --image-id ami-5731123e \
    --launch-permission "Add=[{UserId=123456789012}]"
```

이 명령은 출력을 생성하지 않습니다.

예제 4: AWS 계정에서 시작 권한을 제거하려면

다음 modify-instance-attribute 예제에서는 지정된 AWS 계정에서 시작 권한을 제거합니다.

```
aws ec2 modify-image-attribute \
    --image-id ami-5731123e \
    --launch-permission "Remove=[{UserId=123456789012}]"
```

- 자세한 API 내용은 명령 참조ModifyImageAttribute의 섹션을 참조하세요. AWS CLI

### `modify-instance-attribute`

다음 코드 예시에서는 `modify-instance-attribute`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 인스턴스 유형을 수정하려면

다음 `modify-instance-attribute` 예제에서는 지정된 인스턴스의 인스턴스 유형을 수정합니다. 인스턴스는 `stopped` 상태여야 합니다.

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --instance-type "{\"Value\": \"m1.small\"}"
```

이 명령은 출력을 생성하지 않습니다.

예제 2: 인스턴스에서 향상된 네트워킹 활성화

다음 `modify-instance-attribute` 예제에서는 지정된 인스턴스에 대해 향상된 네트워킹을 활성화합니다. 인스턴스는 `stopped` 상태여야 합니다.

```
aws ec2 modify-instance-attribute \
    --instance-id i-1234567890abcdef0 \
    --sriov-net-support simple
```

이 명령은 출력을 생성하지 않습니다.

예제 3: sourceDestCheck 속성을 수정하려면

다음 `modify-instance-attribute` 예제에서는 지정된 인스턴스의 `sourceDestCheck` 속성을 로 설정합니다`true`. 인스턴스는 에 있어야 합니다VPC.

```
aws ec2 modify-instance-attribute --instance-id i-1234567890abcdef0 --source-dest-
check "{\"Value\": true}"
```

이 명령은 출력을 생성하지 않습니다.

예제 4: 루트 볼륨의 deleteOnTermination 속성을 수정하려면

다음 modify-instance-attribute 예제에서는 지정된 Amazon EBS지원 인스턴스의 루트 볼륨에 대한 deleteOnTermination 속성을 로 설정합니다false. 기본적으로 이 속성은 루트 볼륨에 true 대한 것입니다.

명령:

```
aws ec2 modify-instance-attribute \
  --instance-id i-1234567890abcdef0 \
  --block-device-mappings "[{\"DeviceName\": \"/dev/sda1\",\"Ebs\":
{\"DeleteOnTermination\":false}}]"
```

이 명령은 출력을 생성하지 않습니다.

예제 5: 인스턴스에 연결된 사용자 데이터를 수정하려면

다음 modify-instance-attribute 예제에서는 지정된 인스턴스에 UserData 대해 파일의 내용을 UserData.txt로 추가합니다.

원본 파일의 내용UserData.txt:

```
#!/bin/bash
yum update -y
service httpd start
chkconfig httpd on
```

파일의 내용은 base64 인코딩되어야 합니다. 첫 번째 명령은 텍스트 파일을 base64로 변환하고 새 파일로 저장합니다.

명령의 Linux/macOS 버전:

```
base64 UserData.txt > UserData.base64.txt
```

이 명령은 출력을 생성하지 않습니다.

명령의 Windows 버전:

```
certutil -encode UserData.txt tmp.b64 && findstr /v /c:- tmp.b64 >
 UserData.base64.txt
```

출력:

```
Input Length = 67
Output Length = 152
CertUtil: -encode command completed successfully.
```

이제 다음 CLI 명령에서 해당 파일을 참조할 수 있습니다.

```
aws ec2 modify-instance-attribute \
    --instance-id=i-09b5a14dbca622e76 \
    --attribute userData --value file://UserData.base64.txt
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 사용 설명서의 사용자 데이터 및 AWS CLI 를 참조하세요. EC2

- 자세한 API 내용은 명령 참조ModifyInstanceAttribute의 섹션을 참조하세요. AWS CLI

### modify-instance-capacity-reservation-attributes

다음 코드 예시에서는 modify-instance-capacity-reservation-attributes을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 인스턴스의 용량 예약 대상 설정을 수정하려면

다음 modify-instance-capacity-reservation-attributes 예제에서는 중지된 인스턴스를 특정 용량 예약을 대상으로 수정합니다.

```
aws ec2 modify-instance-capacity-reservation-attributes \
    --instance-id i-EXAMPLE8765abcd4e \
    --capacity-reservation-specification
 'CapacityReservationTarget={CapacityReservationId= cr-1234abcd56EXAMPLE }'
```

출력:

```
{
    "Return": true
}
```

예제 2: 인스턴스의 용량 예약 대상 설정을 수정하려면

다음 modify-instance-capacity-reservation-attributes 예제에서는 속성(인스턴스 유형, 플랫폼, 가용 영역)이 일치하고 인스턴스 일치 기준이 열려 있는 모든 용량 예약에서 지정된 용량 예약을 시작하도록 대상으로 하는 중지된 인스턴스를 수정합니다.

```
aws ec2 modify-instance-capacity-reservation-attributes \
    --instance-id i-EXAMPLE8765abcd4e \
    --capacity-reservation-specification 'CapacityReservationPreference=open'
```

출력:

```
{
    "Return": true
}
```

자세한 내용은 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 인스턴스 용량 예약 설정 수정을 참조하세요.

- 자세한 API 내용은 명령 참조ModifyInstanceCapacityReservationAttributes의 섹션을 참조하세요. AWS CLI

### modify-instance-credit-specification

다음 코드 예시에서는 modify-instance-credit-specification을 사용하는 방법을 보여 줍니다.

AWS CLI

CPU 인스턴스 사용에 대한 크레딧 옵션을 수정하려면

이 예제에서는 지정된 리전에서 지정된 인스턴스를 CPU 사용하기 위한 크레딧 옵션을 '무제한'으로 수정합니다. 유효한 크레딧 옵션은 “표준” 및 “무제한”입니다.

명령:

```
aws ec2 modify-instance-credit-specification --instance-credit-
specification "InstanceId=i-1234567890abcdef0,CpuCredits=unlimited"
```

출력:

```
{
```

```
  "SuccessfulInstanceCreditSpecifications": [
    {
      "InstanceId": "i-1234567890abcdef0"
    }
  ],
  "UnsuccessfulInstanceCreditSpecifications": []
}
```

- 자세한 API 내용은 명령 참조ModifyInstanceCreditSpecification의 섹션을 참조하세요. AWS CLI

### `modify-instance-event-start-time`

다음 코드 예시에서는 `modify-instance-event-start-time`을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스의 이벤트 시작 시간을 수정하려면

다음 `modify-instance-event-start-time` 명령은 지정된 인스턴스의 이벤트 시작 시간을 수정하는 방법을 보여줍니다. `--instance-event-id` 파라미터를 사용하여 이벤트 ID를 지정합니다. `--not-before` 파라미터를 사용하여 새 날짜 및 시간을 지정합니다.

```
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
```

출력:

```
"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"
}
```

자세한 내용은 Amazon Elastic Compute Cloud 사용 설명서의 재부팅이 예약된 인스턴스 작업을 참조하세요.

- 자세한 API 내용은 명령 참조ModifyInstanceEventStartTime의 섹션을 참조하세요. AWS CLI

### modify-instance-event-window

다음 코드 예시에서는 modify-instance-event-window을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 이벤트 창의 시간 범위를 수정하려면

다음 modify-instance-event-window 예제에서는 이벤트 창의 시간 범위를 수정합니다. time-range 파라미터를 사용하여 시간 범위를 수정합니다. cron-expression 파라미터를 함께 지정할 수는 없습니다.

```
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
```

출력:

```
{
    "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"
            }
        ]
    }
}
```

이벤트 기간 제약 조건은 Amazon EC2 사용 설명서의 예약된 이벤트 섹션의 고려 사항을 참조하세요.

예제 2: 이벤트 창의 시간 범위 집합을 수정하려면

다음 `modify-instance-event-window` 예제에서는 이벤트 창의 시간 범위를 수정합니다. `time-range` 파라미터를 사용하여 시간 범위를 수정합니다. `cron-expression` 파라미터를 함께 지정할 수는 없습니다.

```
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}]'
```

출력:

```
{
    "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"
            }
        ]
    }
}
```

이벤트 기간 제약 조건은 Amazon EC2 사용 설명서의 예약된 이벤트 섹션의 고려 사항을 참조하세요.

예제 3: 이벤트 창의 cron 표현식을 수정하려면

다음 `modify-instance-event-window` 예제에서는 이벤트 창의 cron 표현식을 수정합니다. `cron-expression` 파라미터터를 지정하여 cron 표현식을 수정합니다. `time-range` 파라미터를 함께 지정할 수는 없습니다.

```
aws ec2 modify-instance-event-window \
    --region us-east-1 \
    --instance-event-window-id iew-0abcdef1234567890 \
    --cron-expression "* 21-23 * * 2,3"
```

출력:

```
{
    "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"
            }
        ]
    }
}
```

이벤트 기간 제약 조건은 Amazon EC2 사용 설명서의 예약된 이벤트 섹션의 고려 사항을 참조하세요.

- 자세한 API 내용은 명령 참조ModifyInstanceEventWindow의 섹션을 참조하세요. AWS CLI

### `modify-instance-maintenance-options`

다음 코드 예시에서는 `modify-instance-maintenance-options`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 인스턴스의 복구 동작을 비활성화하는 방법

다음 `modify-instance-maintenance-options` 예제에서는 실행 중이거나 중지된 인스턴스에 대한 간소화된 자동 복구를 비활성화합니다.

```
aws ec2 modify-instance-maintenance-options \
    --instance-id i-0abcdef1234567890 \
    --auto-recovery disabled
```

출력:

```
{
    "InstanceId": "i-0abcdef1234567890",
```

```
    "AutoRecovery": "disabled"
}
```

자세한 내용은 Linux 인스턴스용 Amazon 사용 설명서의 인스턴스 복구를 참조하세요. EC2

예제 2: 인스턴스의 복구 동작을 기본값으로 설정하려면

다음 modify-instance-maintenance-options 예제에서는 자동 복구 동작을 기본값으로 설정하여 지원되는 인스턴스 유형에 대해 간소화된 자동 복구를 활성화합니다.

```
aws ec2 modify-instance-maintenance-options \
    --instance-id i-0abcdef1234567890 \
    --auto-recovery default
```

출력:

```
{
    "InstanceId": "i-0abcdef1234567890",
    "AutoRecovery": "default"
}
```

자세한 내용은 Linux 인스턴스용 Amazon 사용 설명서의 인스턴스 복구를 참조하세요. EC2

- 자세한 API 내용은 명령 참조ModifyInstanceMaintenanceOptions의 섹션을 참조하세요. AWS CLI

### modify-instance-metadata-options

다음 코드 예시에서는 modify-instance-metadata-options을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 활성화 IMDSv2

다음 modify-instance-metadata-options 예제에서는 지정된 인스턴스IMDSv2에서 의 사용을 구성합니다.

```
aws ec2 modify-instance-metadata-options \
    --instance-id i-1234567898abcdef0 \
    --http-tokens required \
```

```
    --http-endpoint enabled
```

출력:

```
{
    "InstanceId": "i-1234567898abcdef0",
    "InstanceMetadataOptions": {
        "State": "pending",
        "HttpTokens": "required",
        "HttpPutResponseHopLimit": 1,
        "HttpEndpoint": "enabled"
    }
}
```

자세한 내용은 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 인스턴스 메타데이터 및 사용자 데이터를 참조하세요.

예제 2: 인스턴스 메타데이터 비활성화

다음 `modify-instance-metadata-options` 예제에서는 지정된 인스턴스에서 모든 버전의 인스턴스 메타데이터 사용을 비활성화합니다.

```
aws ec2 modify-instance-metadata-options \
    --instance-id i-1234567898abcdef0 \
    --http-endpoint disabled
```

출력:

```
{
    "InstanceId": "i-1234567898abcdef0",
    "InstanceMetadataOptions": {
        "State": "pending",
        "HttpTokens": "required",
        "HttpPutResponseHopLimit": 1,
        "HttpEndpoint": "disabled"
    }
}
```

자세한 내용은 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 인스턴스 메타데이터 및 사용자 데이터를 참조하세요.

예제 3: 인스턴스에 대해 인스턴스 메타데이터 IPv6 엔드포인트를 활성화하려면

다음 modify-instance-metadata-options 예제에서는 인스턴스 메타데이터 서비스의 IPv6 엔드포인트를 켜는 방법을 보여줍니다.

```
aws ec2 modify-instance-metadata-options \
    --instance-id i-1234567898abcdef0 \
    --http-protocol-ipv6 enabled \
    --http-endpoint enabled
```

출력:

```
{
    "InstanceId": "i-1234567898abcdef0",
    "InstanceMetadataOptions": {
        "State": "pending",
        "HttpTokens": "required",
        "HttpPutResponseHopLimit": 1,
        "HttpEndpoint": "enabled",
        HttpProtocolIpv6": "enabled"
    }
}
```

기본적으로 IPv6 엔드포인트는 비활성화되어 있습니다. 인스턴스를 IPv6전용 서브넷으로 시작한 경우에도 마찬가지입니다. 의 IPv6 엔드포인트IMDS는 Nitro 시스템에 구축된 인스턴스에서만 액세스할 수 있습니다. 자세한 내용은 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 인스턴스 메타데이터 및 사용자 데이터를 참조하세요.

- 자세한 API 내용은 명령 참조ModifyInstanceMetadataOptions의 섹션을 참조하세요. AWS CLI

### modify-instance-placement

다음 코드 예시에서는 modify-instance-placement을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 전용 호스트와 인스턴스의 친화성을 제거하려면

다음 modify-instance-placement 예제에서는 전용 호스트와의 인스턴스 친화성을 제거하고 인스턴스 유형을 지원하는 계정의 사용 가능한 모든 전용 호스트에서 인스턴스를 시작할 수 있습니다.

```
aws ec2 modify-instance-placement \
```

```
    --instance-id i-0e6ddf6187EXAMPLE \
    --affinity default
```

출력:

```
{
    "Return": true
}
```

예제 2: 인스턴스와 지정된 전용 호스트 간에 친화성을 설정하려면

다음 `modify-instance-placement` 예제에서는 인스턴스와 전용 호스트 간의 시작 관계를 설정합니다. 인스턴스는 지정된 전용 호스트에서만 실행할 수 있습니다.

```
aws ec2 modify-instance-placement \
    --instance-id i-0e6ddf6187EXAMPLE \
    --affinity host \
    --host-id i-0e6ddf6187EXAMPLE
```

출력:

```
{
    "Return": true
}
```

자세한 내용은 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 인스턴스 테넌시 및 선호도 수정을 참조하세요.

예제 3: 인스턴스를 배치 그룹으로 이동

다음 `modify-instance-placement` 예제에서는 인스턴스를 배치 그룹으로 이동하고, 인스턴스를 중지하고, 인스턴스 배치를 수정한 다음 인스턴스를 다시 시작합니다.

```
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
```

자세한 내용은 Amazon Elastic Compute Cloud 사용 설명서의 인스턴스에 대한 배치 그룹 변경을 참조하세요.

예제 4: 배치 그룹에서 인스턴스를 제거하려면

다음 `modify-instance-placement` 예제에서는 인스턴스를 중지하고 인스턴스 배치를 수정한 다음 인스턴스를 다시 시작하여 배치 그룹에서 인스턴스를 제거합니다. 다음 예제에서는 인스턴스가 배치 그룹에 있지 않음을 나타내기 위해 배치 그룹 이름에 빈 문자열('')을 지정합니다.

인스턴스 중지:

```
aws ec2 stop-instances \
    --instance-ids i-0123a456700123456
```

배치 수정(Windows 명령 프롬프트, Linux 및 macOS):

```
aws ec2 modify-instance-placement \
    --instance-id i-0123a456700123456 \
    --group-name ""
```

배치 수정(Windows PowerShell):

```
aws ec2 modify-instance-placement `
    --instance-id i-0123a456700123456 `
    --group-name """"
```

인스턴스를 다시 시작합니다.

```
aws ec2 start-instances \
    --instance-ids i-0123a456700123456
```

출력:

```
{
    "Return": true
}
```

자세한 내용은 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 인스턴스 테넌시 및 친화도 수정을 참조하세요.

- 자세한 API 내용은 명령 참조ModifyInstancePlacement의 섹션을 참조하세요. AWS CLI

### modify-ipam-pool

다음 코드 예시에서는 modify-ipam-pool을 사용하는 방법을 보여 줍니다.

AWS CLI

IPAM 풀을 수정하려면

다음 modify-ipam-pool 예제에서는 IPAM 풀을 수정합니다.

(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
```

출력:

```
{
    "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"
            }
        ]
    }
}
```

자세한 내용은 Amazon VPC IPAM 사용 설명서의 풀 편집을 참조하세요.

- 자세한 API 내용은 명령 참조ModifyIpamPool의 섹션을 참조하세요. AWS CLI

### `modify-ipam-resource-cidr`

다음 코드 예시에서는 `modify-ipam-resource-cidr`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 CIDR 할당된 를 수정하려면

다음 `modify-ipam-resource-cidr` 예제에서는 리소스 를 수정합니다CIDR.

(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
```

출력:

```
{
    "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"
    }
}
```

리소스 이동에 대한 자세한 내용은 Amazon VPC IPAM 사용 설명서의 범위 CIDRs 간 리소스 이동을 참조하세요.

모니터링 상태 변경에 대한 자세한 내용은 Amazon VPC IPAM 사용 설명서의 리소스 모니터링 상태 변경을 CIDRs 참조하세요.

- 자세한 API 내용은 명령 참조ModifyIpamResourceCidr의 섹션을 참조하세요. AWS CLI

### `modify-ipam-resource-discovery`

다음 코드 예시에서는 `modify-ipam-resource-discovery`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 검색의 운영 리전을 수정하려면

이 예제에서는 리소스 검색의 운영 리전을 수정하려는 IPAM 위임된 관리자입니다.

이 요청을 완료하려면:

기본 리소스 검색은 수정할 수 없으며 리소스 검색의 소유자여야 합니다. 를 통해 얻을 수 있는 리소스 검색 ID가 필요합니다describe-ipam-resource-discoveries.

다음 `modify-ipam-resource-discovery` 예제에서는 AWS 계정의 기본이 아닌 리소스 검색을 수정합니다.

```
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
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon VPC IPAM 사용 설명서의 리소스 검색 작업을 참조하세요.

- 자세한 API 내용은 명령 참조ModifyIpamResourceDiscovery의 섹션을 참조하세요. AWS CLI

### modify-ipam-scope

다음 코드 예시에서는 modify-ipam-scope을 사용하는 방법을 보여 줍니다.

AWS CLI

범위에 대한 설명을 수정하려면

이 시나리오에서는 IPAM 범위에 대한 설명을 수정하려는 IPAM 위임된 관리자입니다.

이 요청을 완료하려면 범위 ID가 필요하며, 이 ID는 로 가져올 수 있습니다describe-ipam-scopes.

다음 modify-ipam-scope 예제에서는 범위에 대한 설명을 업데이트합니다.

```
aws ec2 modify-ipam-scope \
    --ipam-scope-id ipam-scope-0d3539a30b57dcdd1 \
    --description example \
    --region us-east-1
```

출력:

```
{
"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"
    }
}
```

범위에 대한 자세한 내용은 Amazon VPC IPAM 사용 설명서의 IPAM 작동 방식을 참조하세요.

- 자세한 API 내용은 명령 참조ModifyIpamScope의 섹션을 참조하세요. AWS CLI

### modify-ipam

다음 코드 예시에서는 modify-ipam을 사용하는 방법을 보여 줍니다.

AWS CLI

를 수정하려면 IPAM

다음 modify-ipam 예제에서는 운영 리전을 IPAM 추가하여 를 수정합니다.

(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
```

출력:

```
{
    "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"
    }
}
```

- 자세한 API 내용은 명령 참조ModifyIpam의 섹션을 참조하세요. AWS CLI

### `modify-launch-template`

다음 코드 예시에서는 `modify-launch-template`을 사용하는 방법을 보여 줍니다.

AWS CLI

기본 시작 템플릿 버전을 변경하려면

이 예제에서는 지정된 시작 템플릿의 버전 2를 기본 버전으로 지정합니다.

명령:

```
aws ec2 modify-launch-template --launch-template-id lt-0abcd290751193123 --default-
version 2
```

출력:

```
{
  "LaunchTemplate": {
      "LatestVersionNumber": 2,
      "LaunchTemplateId": "lt-0abcd290751193123",
```

```
        "LaunchTemplateName": "WebServers",
        "DefaultVersionNumber": 2,
        "CreatedBy": "arn:aws:iam::123456789012:root",
        "CreateTime": "2017-12-01T13:35:46.000Z"
    }
}
```

- 자세한 API 내용은 명령 참조ModifyLaunchTemplate의 섹션을 참조하세요. AWS CLI

### `modify-managed-prefix-list`

다음 코드 예시에서는 `modify-managed-prefix-list`을 사용하는 방법을 보여 줍니다.

AWS CLI

접두사 목록을 수정하려면

다음 `modify-managed-prefix-list` 예제에서는 지정된 접두사 목록에 항목을 추가합니다.

```
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
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon VPC 사용 설명서의 관리형 접두사 목록을 참조하세요.

- 자세한 API 내용은 명령 참조ModifyManagedPrefixList의 섹션을 참조하세요. AWS CLI

### `modify-network-interface-attribute`

다음 코드 예시에서는 `modify-network-interface-attribute`을 사용하는 방법을 보여 줍니다.

AWS CLI

네트워크 인터페이스의 연결 속성을 수정하려면

이 예제 명령은 지정된 네트워크 인터페이스의 `attachment` 속성을 수정합니다.

명령:

```
aws ec2 modify-network-interface-attribute --network-interface-id eni-686ea200 --
attachment AttachmentId=eni-attach-43348162,DeleteOnTermination=false
```

네트워크 인터페이스의 설명 속성을 수정하려면

이 예제 명령은 지정된 네트워크 인터페이스의 `description` 속성을 수정합니다.

명령:

```
aws ec2 modify-network-interface-attribute --network-interface-id eni-686ea200 --
description "My description"
```

네트워크 인터페이스의 groupSet 속성을 수정하려면

이 예제 명령은 지정된 네트워크 인터페이스의 `groupSet` 속성을 수정합니다.

명령:

```
aws ec2 modify-network-interface-attribute --network-interface-id eni-686ea200 --
groups sg-903004f8 sg-1a2b3c4d
```

네트워크 인터페이스의 sourceDestCheck 속성을 수정하려면

이 예제 명령은 지정된 네트워크 인터페이스의 `sourceDestCheck` 속성을 수정합니다.

명령:

```
aws ec2 modify-network-interface-attribute --network-interface-id eni-686ea200 --no-
source-dest-check
```

- 자세한 API 내용은 명령 참조ModifyNetworkInterfaceAttribute의 섹션을 참조하세요. AWS CLI

### modify-private-dns-name-options

다음 코드 예시에서는 modify-private-dns-name-options을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스 호스트 이름에 대한 옵션을 수정하려면

다음 modify-private-dns-name-options 예제에서는 DNS A 레코드가 있는 인스턴스 호스트 이름에 대한 DNS 쿼리에 응답하는 옵션을 비활성화합니다.

```
aws ec2 modify-private-dns-name-options \
    --instance-id i-1234567890abcdef0 \
    --no-enable-resource-name-dns-a-record
```

출력:

```
{
    "Return": true
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon EC2 인스턴스 호스트 이름 유형을 참조하세요. EC2

- 자세한 API 내용은 명령 참조ModifyPrivateDnsNameOptions의 섹션을 참조하세요. AWS CLI

### modify-reserved-instances

다음 코드 예시에서는 modify-reserved-instances을 사용하는 방법을 보여 줍니다.

AWS CLI

예약 인스턴스를 수정하려면

이 예제 명령은 예약 인스턴스를 동일한 리전의 다른 가용 영역으로 이동합니다.

명령:

```
aws ec2 modify-reserved-instances --reserved-instances-ids b847fa93-e282-4f55-b59a-1342f5bd7c02 --target-configurations AvailabilityZone=us-west-1c,Platform=EC2-Classic,InstanceCount=10
```

출력:

```
{
  "ReservedInstancesModificationId": "rimod-d3ed4335-b1d3-4de6-ab31-0f13aaf46687"
}
```

예약 인스턴스의 네트워크 플랫폼을 수정하려면

이 예제 명령은 EC2-Classic Reserved Instances를 EC2-로 변환합니다VPC.

명령:

```
aws ec2 modify-reserved-instances --reserved-instances-ids f127bd27-edb7-44c9-a0eb-0d7e09259af0 --target-configurations AvailabilityZone=us-west-1c,Platform=EC2-VPC,InstanceCount=5
```

출력:

```
{
  "ReservedInstancesModificationId": "rimod-82fa9020-668f-4fb6-945d-61537009d291"
}
```

자세한 내용은 Amazon EC2 사용 설명서의 예약 인스턴스 수정을 참조하세요.

예약 인스턴스의 인스턴스 크기를 수정하려면

이 예제 명령은 us-west-1c에 10m1.small Linux/UNIX 인스턴스가 있는 예약 인스턴스를 수정하여 8m1.small 인스턴스가 2m1.large 인스턴스가 되고 나머지 2m1.small 인스턴스는 동일한 가용 영역에서 1m1.medium 인스턴스가 됩니다. 명령:

```
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
```

출력:

```
{
    "ReservedInstancesModificationId": "rimod-acc5f240-080d-4717-b3e3-1c6b11fa00b6"
}
```

자세한 내용은 Amazon EC2 사용 설명서의 예약의 인스턴스 크기 수정을 참조하세요.

- 자세한 API 내용은 명령 참조ModifyReservedInstances의 섹션을 참조하세요. AWS CLI

### modify-security-group-rules

다음 코드 예시에서는 modify-security-group-rules을 사용하는 방법을 보여 줍니다.

AWS CLI

보안 그룹 규칙을 수정하여 규칙 설명, IP 프로토콜 및 CidrIpv4개 주소 범위를 업데이트하려면

다음 modify-security-group-rules 예제에서는 지정된 보안 그룹 규칙의 설명, IP 프로토콜 및 IPV4 CIDR 범위를 업데이트합니다. security-group-rules 파라미터를 사용하여 지정된 보안 그룹 규칙에 대한 업데이트를 입력합니다. 는 모든 프로토콜을 -1 지정합니다.

```
aws ec2 modify-security-group-rules \
    --group-id sg-1234567890abcdef0 \
    --security-group-rules SecurityGroupRuleId=sgr-
abcdef01234567890,SecurityGroupRule='{Description=test,IpProtocol=-1,CidrIpv4=0.0.0.0/0}'
```

출력:

```
{
    "Return": true
}
```

보안 그룹 규칙에 대한 자세한 내용은 Amazon EC2 사용 설명서의 보안 그룹 규칙을 참조하세요.

- 자세한 API 내용은 명령 참조ModifySecurityGroupRules의 섹션을 참조하세요. AWS CLI

### modify-snapshot-attribute

다음 코드 예시에서는 modify-snapshot-attribute을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 스냅샷 속성을 수정하려면

다음 modify-snapshot-attribute 예제에서는 지정된 스냅샷의 createVolumePermission 속성을 업데이트하여 지정된 사용자의 볼륨 권한을 제거합니다.

```
aws ec2 modify-snapshot-attribute \
    --snapshot-id snap-1234567890abcdef0 \
    --attribute createVolumePermission \
    --operation-type remove \
    --user-ids 123456789012
```

예제 2: 스냅샷을 공개하려면

다음 modify-snapshot-attribute 예제에서는 지정된 스냅샷을 공개합니다.

```
aws ec2 modify-snapshot-attribute \
    --snapshot-id snap-1234567890abcdef0 \
    --attribute createVolumePermission \
    --operation-type add \
    --group-names all
```

- 자세한 API 내용은 명령 참조ModifySnapshotAttribute의 섹션을 참조하세요. AWS CLI

### modify-snapshot-tier

다음 코드 예시에서는 modify-snapshot-tier을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 스냅샷 아카이브

다음 modify-snapshot-tier 예제에서는 지정된 스냅샷을 보관합니다.

```
aws ec2 modify-snapshot-tier \
    --snapshot-id snap-01234567890abcedf \
    --storage-tier archive
```

출력:

```
{
    "SnapshotId": "snap-01234567890abcedf",
    "TieringStartTime": "2021-09-15T16:44:37.574Z"
}
```

TieringStartTime 응답 파라미터는 아카이브 프로세스가 시작된 날짜와 시간을 UTC 시간 형식(YYYY-MM-DDTHH:MM:SSZ)으로 나타냅니다.

스냅샷 아카이브에 대한 자세한 내용은 Amazon 사용 설명서의 Amazon EBS 스냅샷 아카이브를 참조하세요. EC2

- 자세한 API 내용은 명령 참조ModifySnapshotTier의 섹션을 참조하세요. AWS CLI

### `modify-spot-fleet-request`

다음 코드 예시에서는 `modify-spot-fleet-request`을 사용하는 방법을 보여 줍니다.

AWS CLI

스팟 플릿 요청을 수정하려면

이 예제 명령은 지정된 스팟 플릿 요청의 대상 용량을 업데이트합니다.

명령:

```
aws ec2 modify-spot-fleet-request --target-capacity 20 --spot-fleet-request-
id sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE
```

출력:

```
{
    "Return": true
}
```

이 예제 명령은 결과적으로 스팟 인스턴스를 종료하지 않고 지정된 스팟 플릿 요청의 대상 용량을 줄입니다.

명령:

```
aws ec2 modify-spot-fleet-request --target-capacity 10 --excess-capacity-
termination-policy NoTermination --spot-fleet-request-ids sfr-73fbd2ce-
aa30-494c-8788-1cee4EXAMPLE
```

출력:

```
{
    "Return": true
}
```

- 자세한 API 내용은 명령 참조ModifySpotFleetRequest의 섹션을 참조하세요. AWS CLI

### `modify-subnet-attribute`

다음 코드 예시에서는 `modify-subnet-attribute`을 사용하는 방법을 보여 줍니다.

AWS CLI

서브넷의 퍼블릭 IPv4 주소 지정 동작을 변경하려면

이 예제에서는 서브넷-1a2b3c4d를 수정하여 이 서브넷으로 시작된 모든 인스턴스에 퍼블릭 IPv4 주소가 할당되도록 지정합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 modify-subnet-attribute --subnet-id subnet-1a2b3c4d --map-public-ip-on-
launch
```

서브넷의 IPv6 주소 지정 동작을 변경하려면

이 예제에서는 서브넷-1a2b3c4d를 수정하여 이 서브넷으로 시작된 모든 인스턴스에 서브넷 범위의 IPv6 주소가 할당되도록 지정합니다.

명령:

```
aws ec2 modify-subnet-attribute --subnet-id subnet-1a2b3c4d --assign-ipv6-address-
on-creation
```

자세한 내용은 Virtual Private Cloud 사용 설명서VPC의 에서 IP 주소 지정을 참조하세요. AWS

- 자세한 API 내용은 명령 참조ModifySubnetAttribute의 섹션을 참조하세요. AWS CLI

### `modify-traffic-mirror-filter-network-services`

다음 코드 예시에서는 `modify-traffic-mirror-filter-network-services`을 사용하는 방법을 보여 줍니다.

AWS CLI

트래픽 미러 필터에 네트워크 서비스를 추가하려면

다음 modify-traffic-mirror-filter-network-services 예제에서는 Amazon DNS 네트워크 서비스를 지정된 필터에 추가합니다.

```
aws ec2 modify-traffic-mirror-filter-network-services \
    --traffic-mirror-filter-id tmf-04812ff784EXAMPLE \
    --add-network-service amazon-dns
```

출력:

```
{
    "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"
            }
        ]
    }
}
```

자세한 내용은 트래픽 미러링 가이드의 트래픽 미러 필터 네트워크 서비스 수정을 참조하세요. AWS

- 자세한 API 내용은 명령 참조ModifyTrafficMirrorFilterNetworkServices의 섹션을 참조하세요. AWS CLI

### modify-traffic-mirror-filter-rule

다음 코드 예시에서는 modify-traffic-mirror-filter-rule을 사용하는 방법을 보여 줍니다.

AWS CLI

트래픽 미러 필터 규칙을 수정하려면

다음 modify-traffic-mirror-filter-rule 예제에서는 지정된 트래픽 미러 필터 규칙에 대한 설명을 수정합니다.

```
aws ec2 modify-traffic-mirror-filter-rule \
    --traffic-mirror-filter-rule-id tmfr-0ca76e0e08EXAMPLE \
    --description "TCP Rule"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 트래픽 미러링 가이드의 트래픽 미러 필터 규칙 수정을 참조하세요. AWS

- 자세한 API 내용은 명령 참조ModifyTrafficMirrorFilterRule의 섹션을 참조하세요. AWS CLI

### modify-traffic-mirror-session

다음 코드 예시에서는 modify-traffic-mirror-session을 사용하는 방법을 보여 줍니다.

AWS CLI

트래픽 미러 세션을 수정하려면

다음 modify-traffic-mirror-session 예제에서는 트래픽 미러 세션 설명과 미러링할 패킷 수를 변경합니다.

```
aws ec2 modify-traffic-mirror-session \
    --description "Change packet length" \
    --traffic-mirror-session-id tms-08a33b1214EXAMPLE \
    --remove-fields "packet-length"
```

출력:

```
{
    "TrafficMirrorSession": {
        "TrafficMirrorSessionId": "tms-08a33b1214EXAMPLE",
        "TrafficMirrorTargetId": "tmt-07f75d8feeEXAMPLE",
        "TrafficMirrorFilterId": "tmf-04812ff784EXAMPLE",
        "NetworkInterfaceId": "eni-070203f901EXAMPLE",
        "OwnerId": "111122223333",
        "SessionNumber": 1,
        "VirtualNetworkId": 7159709,
        "Description": "Change packet length",
        "Tags": []
    }
}
```

자세한 내용은 트래픽 미러링 가이드의 트래픽 미러 세션 수정을 참조하세요.

- 자세한 API 내용은 명령 참조ModifyTrafficMirrorSession의 섹션을 참조하세요. AWS CLI

### modify-transit-gateway-prefix-list-reference

다음 코드 예시에서는 modify-transit-gateway-prefix-list-reference을 사용하는 방법을 보여 줍니다.

AWS CLI

접두사 목록에 대한 참조를 수정하려면

다음 modify-transit-gateway-prefix-list-reference 예제에서는 트래픽이 라우팅되는 첨부 파일을 변경하여 지정된 라우팅 테이블의 접두사 목록 참조를 수정합니다.

```
aws ec2 modify-transit-gateway-prefix-list-reference \
```

```
    --transit-gateway-route-table-id tgw-rtb-0123456789abcd123 \
    --prefix-list-id pl-1111122222222333 \
    --transit-gateway-attachment-id tgw-attach-aabbccddaabbccaab
```

출력:

```
{
    "TransitGatewayPrefixListReference": {
        "TransitGatewayRouteTableId": "tgw-rtb-0123456789abcd123",
        "PrefixListId": "pl-11111122222222333",
        "PrefixListOwnerId": "123456789012",
        "State": "modifying",
        "Blackhole": false,
        "TransitGatewayAttachment": {
            "TransitGatewayAttachmentId": "tgw-attach-aabbccddaabbccaab",
            "ResourceType": "vpc",
            "ResourceId": "vpc-112233445566aabbc"
        }
    }
}
```

자세한 내용은 Transit Gateways 가이드의 접두사 목록 참조를 참조하세요.

- 자세한 API 내용은 명령 참조ModifyTransitGatewayPrefixListReference의 섹션을 참조하세요. AWS CLI

### `modify-transit-gateway-vpc-attachment`

다음 코드 예시에서는 `modify-transit-gateway-vpc-attachment`을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 VPC 연결을 수정하려면

다음 `modify-transit-gateway-vpc-attachment` 예제에서는 지정된 전송 게이트웨이 VPC 연결에 서브넷을 추가합니다.

```
aws ec2 modify-transit-gateway-vpc-attachment \
    --transit-gateway-attachment-id tgw-attach-09fbd47ddfEXAMPLE \
    --add-subnet-ids subnet-0e51f45802EXAMPLE
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 Transit Gateways Guide의 Transit Gateway attachments to VPC a를 참조하세요.

- 자세한 API 내용은 명령 참조ModifyTransitGatewayVpcAttachment의 섹션을 참조하세요. AWS CLI

### `modify-transit-gateway`

다음 코드 예시에서는 `modify-transit-gateway`을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이를 수정하려면

다음 `modify-transit-gateway` 예제에서는 VPN 첨부 파일 ECMP 지원을 활성화하여 지정된 전송 게이트웨이를 수정합니다.

```
aws ec2 modify-transit-gateway \
    --transit-gateway-id tgw-111111222222aaaaa \
    --options VpnEcmpSupport=enable
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 Transit Gateways 가이드의 Transit Gateways를 참조하세요.

- 자세한 API 내용은 명령 참조ModifyTransitGateway의 섹션을 참조하세요. AWS CLI

### modify-verified-access-endpoint-policy

다음 코드 예시에서는 modify-verified-access-endpoint-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

엔드포인트에 대한 Verified Access 정책을 구성하려면

다음 modify-verified-access-endpoint-policy 예제에서는 지정된 Verified Access 정책을 지정된 Verified Access 엔드포인트에 추가합니다.

```
aws ec2 modify-verified-access-endpoint-policy \
    --verified-access-endpoint-id vae-066fac616d4d546f2 \
    --policy-enabled \
    --policy-document file://policy.txt
```

policy.txt의 콘텐츠:

```
permit(principal,action,resource)
when {
    context.identity.groups.contains("finance") &&
    context.identity.email.verified == true
};
```

출력:

```
{
    "PolicyEnabled": true,
    "PolicyDocument": "permit(principal,action,resource)\nwhen
 {\n    context.identity.groups.contains(\"finance\") &&\n
 context.identity.email_verified == true\n};"
}
```

자세한 내용은 Verified Access 사용 설명서의 Verified Access 정책을 참조하세요. AWS

- 자세한 API 내용은 명령 참조ModifyVerifiedAccessEndpointPolicy의 섹션을 참조하세요. AWS CLI

### `modify-verified-access-endpoint`

다음 코드 예시에서는 `modify-verified-access-endpoint`을 사용하는 방법을 보여 줍니다.

AWS CLI

Verified Access 엔드포인트의 구성을 수정하려면

다음 `modify-verified-access-endpoint` 예제에서는 지정된 Verified Access 엔드포인트에 지정된 설명을 추가합니다.

```
aws ec2 modify-verified-access-endpoint \
    --verified-access-endpoint-id vae-066fac616d4d546f2 \
    --description "Testing Verified Access"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Verified Access 사용 설명서의 Verified Access 엔드포인트를 참조하세요. AWS

- 자세한 API 내용은 명령 참조ModifyVerifiedAccessEndpoint의 섹션을 참조하세요. AWS CLI

### `modify-verified-access-group-policy`

다음 코드 예시에서는 `modify-verified-access-group-policy`을 사용하는 방법을 보여 줍니다.

AWS CLI

그룹에 대한 Verified Access 정책을 구성하려면

다음 `modify-verified-access-group-policy` 예제에서는 지정된 Verified Access 정책을 지정된 Verified Access 그룹에 추가합니다.

```
aws ec2 modify-verified-access-group-policy \
```

```
    --verified-access-group-id vagr-0dbe967baf14b7235 \
    --policy-enabled \
    --policy-document file://policy.txt
```

policy.txt의 콘텐츠:

```
permit(principal,action,resource)
when {
    context.identity.groups.contains("finance") &&
    context.identity.email.verified == true
};
```

출력:

```
{
    "PolicyEnabled": true,
    "PolicyDocument": "permit(principal,action,resource)\nwhen
 {\n    context.identity.groups.contains(\"finance\") &&\n
 context.identity.email_verified == true\n};"
}
```

자세한 내용은 Verified Access 사용 설명서의 Verified Access 그룹을 참조하세요. AWS

- 자세한 API 내용은 명령 참조ModifyVerifiedAccessGroupPolicy의 섹션을 참조하세요. AWS CLI

### modify-verified-access-group

다음 코드 예시에서는 modify-verified-access-group을 사용하는 방법을 보여 줍니다.

AWS CLI

Verified Access 그룹의 구성을 수정하려면

다음 modify-verified-access-group 예제에서는 지정된 Verified Access 그룹에 지정된 설명을 추가합니다.

```
aws ec2 modify-verified-access-group \
    --verified-access-group-id vagr-0dbe967baf14b7235 \
    --description "Testing Verified Access"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Verified Access 사용 설명서의 Verified Access 그룹을 참조하세요. AWS

- 자세한 API 내용은 명령 참조ModifyVerifiedAccessGroup의 섹션을 참조하세요. AWS CLI

### `modify-verified-access-instance-logging-configuration`

다음 코드 예시에서는 `modify-verified-access-instance-logging-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

Verified Access 인스턴스에 대한 로깅을 활성화하려면

다음 `modify-verified-access-instance-logging-configuration` 예제에서는 지정된 Verified Access 인스턴스에 대한 액세스 로깅을 활성화합니다. 로그는 지정된 CloudWatch 로그 로그 그룹에 전달됩니다.

```
aws ec2 modify-verified-access-instance-logging-configuration \
    --verified-access-instance-id vai-0ce000c0b7643abea \
    --access-logs CloudWatchLogs={Enabled=true,LogGroup=my-log-group}
```

출력:

```
{
    "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
        }
    }
}
```

자세한 내용은 Verified Access 사용 설명서의 Verified Access 로그를 참조하세요. AWS

- 자세한 API 내용은 명령 참조ModifyVerifiedAccessInstanceLoggingConfiguration의 섹션을 참조하세요. AWS CLI

### `modify-verified-access-instance`

다음 코드 예시에서는 `modify-verified-access-instance`을 사용하는 방법을 보여 줍니다.

AWS CLI

Verified Access 인스턴스의 구성을 수정하려면

다음 `modify-verified-access-instance` 예제에서는 지정된 Verified Access 인스턴스에 지정된 설명을 추가합니다.

```
aws ec2 modify-verified-access-instance \
    --verified-access-instance-id vai-0ce000c0b7643abea \
    --description "Testing Verified Access"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Verified Access 사용 설명서의 Verified Access 인스턴스를 참조하세요. AWS

- 자세한 API 내용은 명령 참조ModifyVerifiedAccessInstance의 섹션을 참조하세요. AWS CLI

### `modify-verified-access-trust-provider`

다음 코드 예시에서는 `modify-verified-access-trust-provider`을 사용하는 방법을 보여 줍니다.

AWS CLI

Verified Access 신뢰 공급자의 구성을 수정하려면

다음 `modify-verified-access-trust-provider` 예제에서는 지정된 Verified Access 신뢰 공급자에 지정된 설명을 추가합니다.

```
aws ec2 modify-verified-access-trust-provider \
    --verified-access-trust-provider-id vatp-0bb32de759a3e19e7 \
    --description "Testing Verified Access"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Verified Access 사용 설명서의 Verified Access에 대한 신뢰 공급자를 참조하세요. AWS

- 자세한 API 내용은 명령 참조ModifyVerifiedAccessTrustProvider의 섹션을 참조하세요. AWS CLI

### modify-volume-attribute

다음 코드 예시에서는 modify-volume-attribute을 사용하는 방법을 보여 줍니다.

AWS CLI

볼륨 속성을 수정하려면

이 예제에서는 ID가 인 볼륨의 autoEnableIo 속성을 vol-1234567890abcdef0로 설정합니다true. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 modify-volume-attribute --volume-id vol-1234567890abcdef0 --auto-enable-io
```

- 자세한 API 내용은 명령 참조ModifyVolumeAttribute의 섹션을 참조하세요. AWS CLI

### modify-volume

다음 코드 예시에서는 modify-volume을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 크기를 변경하여 볼륨을 수정하려면

다음 modify-volume 예제에서는 지정된 볼륨의 크기를 150GB 로 변경합니다.

명령:

```
aws ec2 modify-volume --size 150 --volume-id vol-1234567890abcdef0
```

출력:

```
{
    "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
    }
}
```

예제 2: 유형, 크기 및 IOPS 값을 변경하여 볼륨을 수정하려면

다음 `modify-volume` 예제에서는 볼륨 유형을 프로비저닝됨 IOPS 으로 변경하고SSD, 목표 IOPS 속도를 10000으로 설정하고, 볼륨 크기를 350GB 로 설정합니다.

```
aws ec2 modify-volume \
    --volume-type io1 \
    --iops 10000 \
    --size 350 \
    --volume-id vol-1234567890abcdef0
```

출력:

```
{
    "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
```

```
    }
}
```

- 자세한 API 내용은 명령 참조ModifyVolume의 섹션을 참조하세요. AWS CLI

### `modify-vpc-attribute`

다음 코드 예시에서는 `modify-vpc-attribute`을 사용하는 방법을 보여 줍니다.

AWS CLI

enableDnsSupport 속성을 수정하려면

이 예제에서는 `enableDnsSupport` 속성을 수정합니다. 이 속성은 에 대한 DNS 해상도가 활성화되었는지 여부를 나타냅니다VPC. 이 속성이 인 경우 Amazon DNS 서버는 인스턴스`true`의 DNS 호스트 이름을 해당 IP 주소로 확인하지만, 그렇지 않으면 확인하지 않습니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 modify-vpc-attribute --vpc-id vpc-a01106c2 --enable-dns-support "{\"Value
\":false}"
```

enableDnsHostnames 속성을 수정하려면

이 예제에서는 `enableDnsHostnames` 속성을 수정합니다. 이 속성은 인스턴스가 DNS 호스트 이름 VPC 가져오기에서 시작되었는지 여부를 나타냅니다. 이 속성이 `true`인 경우 DNS 호스트 이름 VPC 가져오기의 인스턴스입니다. 그렇지 않으면 인스턴스가 되지 않습니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 modify-vpc-attribute --vpc-id vpc-a01106c2 --enable-dns-hostnames "{\"Value
\":false}"
```

- 자세한 API 내용은 명령 참조ModifyVpcAttribute의 섹션을 참조하세요. AWS CLI

### `modify-vpc-endpoint-connection-notification`

다음 코드 예시에서는 `modify-vpc-endpoint-connection-notification`을 사용하는 방법을 보여 줍니다.

AWS CLI

엔드포인트 연결 알림을 수정하려면

이 예제에서는 지정된 엔드포인트 연결 알림에 대한 SNS 주제를 변경합니다.

명령:

```
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
```

출력:

```
{
   "ReturnValue": true
}
```

- 자세한 API 내용은 명령 참조ModifyVpcEndpointConnectionNotification의 섹션을 참조하세요. AWS CLI

### modify-vpc-endpoint-service-configuration

다음 코드 예시에서는 modify-vpc-endpoint-service-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

엔드포인트 서비스 구성을 수정하려면

이 예제에서는 지정된 엔드포인트 서비스에 대한 수락 요구 사항을 변경합니다.

명령:

```
aws ec2 modify-vpc-endpoint-service-configuration --service-id vpce-
svc-09222513e6e77dc86 --no-acceptance-required
```

출력:

```
{
   "ReturnValue": true
```

```
}
```

- 자세한 API 내용은 명령 참조ModifyVpcEndpointServiceConfiguration의 섹션을 참조하세요. AWS CLI

### modify-vpc-endpoint-service-payer-responsibility

다음 코드 예시에서는 modify-vpc-endpoint-service-payer-responsibility을 사용하는 방법을 보여 줍니다.

AWS CLI

지불자 책임을 수정하려면

다음 modify-vpc-endpoint-service-payer-responsibility 예제에서는 지정된 엔드포인트 서비스의 지불자 책임을 수정합니다.

```
aws ec2 modify-vpc-endpoint-service-payer-responsibility \
    --service-id vpce-svc-071afff70666e61e0 \
    --payer-responsibility ServiceOwner
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조ModifyVpcEndpointServicePayerResponsibility의 섹션을 참조하세요. AWS CLI

### modify-vpc-endpoint-service-permissions

다음 코드 예시에서는 modify-vpc-endpoint-service-permissions을 사용하는 방법을 보여 줍니다.

AWS CLI

엔드포인트 서비스 권한을 수정하려면

이 예제에서는 AWS 계정이 지정된 엔드포인트 서비스에 연결할 수 있는 권한을 추가합니다.

명령:

```
aws ec2 modify-vpc-endpoint-service-permissions --service-id vpce-
svc-03d5ebb7d9579a2b3 --add-allowed-principals '["arn:aws:iam::123456789012:root"]'
```

출력:

```
{
   "ReturnValue": true
}
```

이 예제에서는 특정 IAM 사용자(admin)가 지정된 엔드포인트 서비스에 연결할 수 있는 권한을 추가합니다.

명령:

```
aws ec2 modify-vpc-endpoint-service-permissions --service-id vpce-
svc-03d5ebb7d9579a2b3 --add-allowed-principals '["arn:aws:iam::123456789012:user/
admin"]'
```

- 자세한 API 내용은 명령 참조ModifyVpcEndpointServicePermissions의 섹션을 참조하세요. AWS CLI

### `modify-vpc-endpoint`

다음 코드 예시에서는 `modify-vpc-endpoint`을 사용하는 방법을 보여 줍니다.

AWS CLI

게이트웨이 엔드포인트를 수정하려면

이 예제에서는 라우팅 테이블`rtb-aaa222bb`을 엔드포인트와 연결하고 정책 문서를 재설정`vpce-1a2b3c4d`하여 게이트웨이 엔드포인트를 수정합니다.

명령:

```
aws ec2 modify-vpc-endpoint --vpc-endpoint-id vpce-1a2b3c4d --add-route-table-
ids rtb-aaa222bb --reset-policy
```

출력:

```
{
  "Return": true
}
```

인터페이스 엔드포인트를 수정하려면

이 예제에서는 엔드포인트vpce-0fe5b17a0707d6fa5에 서브넷subnet-d6fcaa8d을 추가하여 인터페이스 엔드포인트를 수정합니다.

명령:

```
aws ec2 modify-vpc-endpoint --vpc-endpoint-id vpce-0fe5b17a0707d6fa5 --add-subnet-
id subnet-d6fcaa8d
```

출력:

```
{
  "Return": true
}
```

- 자세한 API 내용은 명령 참조ModifyVpcEndpoint의 섹션을 참조하세요. AWS CLI

### modify-vpc-peering-connection-options

다음 코드 예시에서는 modify-vpc-peering-connection-options을 사용하는 방법을 보여 줍니다.

AWS CLI

로컬 연결에서 VPC 피어링 ClassicLink 연결을 통한 통신을 활성화하려면

이 예제에서는 피어링 연결 의 경우 요청자의 pcx-aaaabbb소유자가 VPC 피어링 연결 옵션을 VPC 수정하여 로컬 ClassicLink 연결이 피어 와 통신할 수 있도록 합니다VPC.

명령:

```
aws ec2 modify-vpc-peering-connection-options --vpc-peering-
connection-id pcx-aaaabbbb --requester-peering-connection-
options AllowEgressFromLocalClassicLinkToRemoteVpc=true
```

출력:

```
{
```

```
  "RequesterPeeringConnectionOptions": {
      "AllowEgressFromLocalClassicLinkToRemoteVpc": true
  }
}
```

로컬에서 VPC 원격 연결로의 VPC 피어링 ClassicLink 연결을 통한 통신을 활성화하려면

이 예제에서는 수락자의 소유자가 VPC 피어링 연결 옵션을 VPC 수정하여 로컬이 피어 의 ClassicLink 연결과 통신VPC할 수 있도록 합니다VPC.

명령:

```
aws ec2 modify-vpc-peering-connection-options --vpc-peering-
connection-id pcx-aaaabbbb --accepter-peering-connection-
options AllowEgressFromLocalVpcToRemoteClassicLink=true
```

출력:

```
{
  "AccepterPeeringConnectionOptions": {
    "AllowEgressFromLocalVpcToRemoteClassicLink": true
  }
}
```

VPC 피어링 연결에 대한 DNS 해결 지원을 활성화하려면

이 예제에서는 요청자의 소유자가 VPC 에 대한 VPC 피어링 연결 옵션을 수정pcx-aaaabbbb하여 피어 의 인스턴스에서 쿼리할 때 로컬이 퍼블릭 DNS 호스트 이름을 프라이빗 IP 주소로 VPC 확인할 수 있도록 합니다VPC.

명령:

```
aws ec2 modify-vpc-peering-connection-options --vpc-peering-connection-id pcx-
aaaabbbb --requester-peering-connection-options AllowDnsResolutionFromRemoteVpc=true
```

출력:

```
{
  "RequesterPeeringConnectionOptions": {
```

```
        "AllowDnsResolutionFromRemoteVpc": true
    }
}
```

- 자세한 API 내용은 명령 참조ModifyVpcPeeringConnectionOptions의 섹션을 참조하세요. AWS CLI

### `modify-vpc-tenancy`

다음 코드 예시에서는 `modify-vpc-tenancy`을 사용하는 방법을 보여 줍니다.

AWS CLI

의 테넌시를 수정하려면 VPC

이 예제에서는 의 테넌시를 VPC`vpc-1a2b3c4d`로 수정합니다`default`.

명령:

```
aws ec2 modify-vpc-tenancy --vpc-id vpc-1a2b3c4d --instance-tenancy default
```

출력:

```
{
    "Return": true
}
```

- 자세한 API 내용은 명령 참조ModifyVpcTenancy의 섹션을 참조하세요. AWS CLI

### `modify-vpn-connection-options`

다음 코드 예시에서는 `modify-vpn-connection-options`을 사용하는 방법을 보여 줍니다.

AWS CLI

VPN 연결 옵션을 수정하려면

다음 `modify-vpn-connection-options` 예제에서는 지정된 VPN 연결의 고객 게이트웨이 측 IPv4CIDR에 있는 로컬을 수정합니다.

```
aws ec2 modify-vpn-connection-options \
    --vpn-connection-id vpn-1122334455aabbccd \
    --local-ipv4-network-cidr 10.0.0.0/16
```

출력:

```
{
    "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": ""
                }
            ]
        }
    ]
}
```

자세한 내용은 AWS Site-to-Site VPN 사용 설명서의 VPN 연결 옵션 수정을 참조하세요 Site-to-Site.

- 자세한 API 내용은 명령 참조ModifyVpnConnectionOptions의 섹션을 참조하세요. AWS CLI

### `modify-vpn-connection`

다음 코드 예시에서는 `modify-vpn-connection`을 사용하는 방법을 보여 줍니다.

AWS CLI

VPN 연결을 수정하려면

다음 `modify-vpn-connection` 예제에서는 가상 프라이빗 게이트웨이VPN에 연결vpn-12345678901234567하기 위한 대상 게이트웨이를 변경합니다vgw-11223344556677889.

```
aws ec2 modify-vpn-connection \
    --vpn-connection-id vpn-12345678901234567 \
    --vpn-gateway-id vgw-11223344556677889
```

출력:

```
{
    "VpnConnection": {
        "CustomerGatewayConfiguration": "...configuration information...",
        "CustomerGatewayId": "cgw-aabbccddee1122334",
        "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"
            }
        ]
    }
}
```

- 자세한 API 내용은 명령 참조ModifyVpnConnection의 섹션을 참조하세요. AWS CLI

### `modify-vpn-tunnel-certificate`

다음 코드 예시에서는 `modify-vpn-tunnel-certificate`을 사용하는 방법을 보여 줍니다.

AWS CLI

VPN 터널 인증서를 교체하려면

다음 `modify-vpn-tunnel-certificate` 예제에서는 VPN 연결을 위해 지정된 터널의 인증서를 교체합니다.

```
aws ec2 modify-vpn-tunnel-certificate \
    --vpn-tunnel-outside-ip-address 203.0.113.17 \
    --vpn-connection-id vpn-12345678901234567
```

출력:

```
{
    "VpnConnection": {
        "CustomerGatewayConfiguration": ...configuration information...,
        "CustomerGatewayId": "cgw-aabbccddee1122334",
```

```
        "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"
            }
        ]
    }
}
```

- 자세한 API 내용은 명령 참조ModifyVpnTunnelCertificate의 섹션을 참조하세요. AWS CLI

### `modify-vpn-tunnel-options`

다음 코드 예시에서는 `modify-vpn-tunnel-options`을 사용하는 방법을 보여 줍니다.

AWS CLI

VPN 연결에 대한 터널 옵션을 수정하려면

다음 `modify-vpn-tunnel-options` 예제에서는 지정된 터널 및 VPN 연결에 허용되는 Diffie-Hellman 그룹을 업데이트합니다.

```
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}]
```

출력:

```
{
    "VpnConnection": {
        "CustomerGatewayConfiguration": "...configuration information...",
        "CustomerGatewayId": "cgw-aabbccddee1122334",
        "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"
            }
        ]
    }
}
```

- 자세한 API 내용은 명령 참조ModifyVpnTunnelOptions의 섹션을 참조하세요. AWS CLI

### monitor-instances

다음 코드 예시에서는 monitor-instances을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스에 대한 세부 모니터링을 활성화하는 방법

이 예제 명령은 지정된 인스턴스에 대한 세부 모니터링을 활성화합니다.

명령:

```
aws ec2 monitor-instances --instance-ids i-1234567890abcdef0
```

출력:

```
{
  "InstanceMonitorings": [
      {
          "InstanceId": "i-1234567890abcdef0",
          "Monitoring": {
              "State": "pending"
          }
      }
  ]
}
```

- 자세한 API 내용은 명령 참조MonitorInstances의 섹션을 참조하세요. AWS CLI

### move-address-to-vpc

다음 코드 예시에서는 move-address-to-vpc을 사용하는 방법을 보여 줍니다.

AWS CLI

주소를 EC2-로 이동하려면VPC

이 예제에서는 탄력적 IP 주소 54.123.4.56을 EC2-VPC 플랫폼으로 이동합니다.

명령:

```
aws ec2 move-address-to-vpc --public-ip 54.123.4.56
```

출력:

```
{
  "Status": "MoveInProgress"
}
```

- 자세한 API 내용은 명령 참조MoveAddressToVpc의 섹션을 참조하세요. AWS CLI

### move-byoip-cidr-to-ipam

다음 코드 예시에서는 move-byoip-cidr-to-ipam을 사용하는 방법을 보여 줍니다.

AWS CLI

를 BYOIP CIDR 로 전송하려면 IPAM

다음 move-byoip-cidr-to-ipam 예제에서는 를 BYOIPCIDR로 전송합니다IPAM.

(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
```

출력:

```
{
    "ByoipCidr": {
        "Cidr": "130.137.249.0/24",
        "State": "pending-transfer"
    }
}
```

자세한 내용은 Amazon VPC IPAM 사용 설명서의 자습서: 기존 을 BYOIPIPv4CIDR로 전송IPAM을 참조하세요.

- 자세한 API 내용은 명령 참조MoveByoipCidrToIpam의 섹션을 참조하세요. AWS CLI

### network-insights-access-scope

다음 코드 예시에서는 network-insights-access-scope을 사용하는 방법을 보여 줍니다.

AWS CLI

Network Insights 액세스 범위를 생성하려면

다음 create-network-insights-access-scope 예제에서는 AWS 계정에 네트워크 인사이트 액세스 범위를 생성합니다.

```
aws ec2 create-network-insights-access-scope \
    --cli-input-json file://access-scope-file.json
```

access-scope-file.json의 콘텐츠:

```
{
    {
        "MatchPaths": [
            {
                "Source": {
                    "ResourceStatement": {
                         "Resources": [
                            "vpc-abcd12e3"
                        ]
                    }
                }
            }
        ],
        "ExcludePaths": [
            {
                "Source": {
                    "ResourceStatement": {
                        "ResourceTypes": [
                            "AWS::EC2::InternetGateway"
                        ]
                    }
                }
```

```
            }
        ]
    }
}
```

출력:

```
{
    "NetworkInsightsAccessScopeAnalysisId": "nisa-123456789111"
    }{
    "NetworkInsightsAccessScope": {
        "NetworkInsightsAccessScopeId": "nis-123456789222",
        "NetworkInsightsAccessScopeArn": "arn:aws:ec2:us-
east-1:123456789222:network-insights-access-scope/nis-123456789222",
        "CreatedDate": "2022-01-25T19:20:28.796000+00:00",
        "UpdatedDate": "2022-01-25T19:20:28.797000+00:00"
    },
    "NetworkInsightsAccessScopeContent": {
        "NetworkInsightsAccessScopeId": "nis-04c0c0fbca737c404",
        "MatchPaths": [
            {
                "Source": {
                    "ResourceStatement": {
                        "Resources": [
                            "vpc-abcd12e3"
                        ]
                    }
                }
            }
        ],
        "ExcludePaths": [
            {
                "Source": {
                    "ResourceStatement": {
                        "ResourceTypes": [
                            "AWS::EC2::InternetGateway"
                        ]
                    }
                }
            }
        ]
    }
}
```

자세한 내용은 Network Access Analyzer 가이드의 를 사용하여 AWS CLI Network Access Analyzer 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조NetworkInsightsAccessScope의 섹션을 참조하세요. AWS CLI

### `provision-byoip-cidr`

다음 코드 예시에서는 `provision-byoip-cidr`을 사용하는 방법을 보여 줍니다.

AWS CLI

주소 범위를 프로비저닝하려면

다음 `provision-byoip-cidr` 예제에서는 와 함께 사용할 퍼블릭 IP 주소 범위를 프로비저닝합니다 AWS.

```
aws ec2 provision-byoip-cidr \
    --cidr 203.0.113.25/24 \
    --cidr-authorization-context Message="$text_message",Signature="$signed_message"
```

출력:

```
{
    "ByoipCidr": {
        "Cidr": "203.0.113.25/24",
        "State": "pending-provision"
    }
}
```

권한 부여 컨텍스트에 대한 메시지 문자열 생성에 대한 자세한 내용은 Amazon EC2 사용 설명서의 고유 IP 주소 가져오기를 참조하세요.

- 자세한 API 내용은 명령 참조ProvisionByoipCidr의 섹션을 참조하세요. AWS CLI

### `provision-ipam-pool-cidr`

다음 코드 예시에서는 `provision-ipam-pool-cidr`을 사용하는 방법을 보여 줍니다.

AWS CLI

IPAM 풀CIDR에 를 프로비저닝하려면

다음 provision-ipam-pool-cidr 예제에서는 IPAM 풀CIDR에 를 프로비저닝합니다.

(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
```

출력:

```
{
    "IpamPoolCidr": {
        "Cidr": "10.0.0.0/24",
        "State": "pending-provision"
    }
}
```

자세한 내용은 Amazon VPC IPAM 사용 설명서의 풀CIDRs에 프로비저닝을 참조하세요.

- 자세한 API 내용은 명령 참조ProvisionIpamPoolCidr의 섹션을 참조하세요. AWS CLI

### purchase-host-reservation

다음 코드 예시에서는 purchase-host-reservation을 사용하는 방법을 보여 줍니다.

AWS CLI

전용 호스트 예약을 구매하려면

이 예제에서는 계정의 지정된 전용 호스트에 대해 지정된 전용 호스트 예약 제품을 구매합니다.

명령:

```
aws ec2 purchase-host-reservation --offering-id hro-03f707bf363b6b324 --host-id-
set h-013abcd2a00cbd123
```

출력:

```
{
  "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"
}
```

- 자세한 API 내용은 명령 참조PurchaseHostReservation의 섹션을 참조하세요. AWS CLI

### purchase-reserved-instances-offering

다음 코드 예시에서는 purchase-reserved-instances-offering을 사용하는 방법을 보여 줍니다.

AWS CLI

예약 인스턴스 제품을 구매하려면

이 예제 명령은 제공 ID 및 인스턴스 수를 지정하여 예약 인스턴스 제공의 구매를 보여줍니다.

명령:

```
aws ec2 purchase-reserved-instances-offering --reserved-instances-offering-
id ec06327e-dd07-46ee-9398-75b5fexample --instance-count 3
```

출력:

```
{
  "ReservedInstancesId": "af9f760e-6f91-4559-85f7-4980eexample"
```

```
}
```

- 자세한 API 내용은 명령 참조PurchaseReservedInstancesOffering의 섹션을 참조하세요. AWS CLI

### `purchase-scheduled-instances`

다음 코드 예시에서는 `purchase-scheduled-instances`을 사용하는 방법을 보여 줍니다.

AWS CLI

예약된 인스턴스를 구매하려면

이 예제에서는 예약된 인스턴스를 구매합니다.

명령:

```
aws ec2 purchase-scheduled-instances --purchase-requests file://purchase-request.json
```

Purchase-request.json:

```
[
    {
        "PurchaseToken": "eyJ2IjoiMSIsInMiOjEsImMiOi...",
        "InstanceCount": 1
    }
]
```

출력:

```
{
  "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"
    }
  ]
}
```

- 자세한 API 내용은 명령 참조PurchaseScheduledInstances의 섹션을 참조하세요. AWS CLI

### reboot-instances

다음 코드 예시에서는 reboot-instances을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon EC2 인스턴스를 재부팅하려면

이 예제에서는 지정된 인스턴스를 재부팅합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 reboot-instances --instance-ids i-1234567890abcdef5
```

자세한 내용은 Amazon Elastic Compute Cloud 사용 설명서에서 인스턴스 재부팅을 참조하세요.

- 자세한 API 내용은 명령 참조RebootInstances의 섹션을 참조하세요. AWS CLI

### register-image

다음 코드 예시에서는 register-image을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 매니페스트 파일을 AMI 사용하여 를 등록하려면

다음 `register-image` 예제에서는 Amazon S3에서 지정된 매니페스트 파일을 AMI 사용하여 를 등록합니다.

```
aws ec2 register-image \
    --name my-image \
    --image-location my-s3-bucket/myimage/image.manifest.xml
```

출력:

```
{
    "ImageId": "ami-1234567890EXAMPLE"
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon Machine Images(AMI)를 참조하세요. EC2

예제 2: 루트 디바이스의 스냅샷을 AMI 사용하여 를 등록하려면

다음 `register-image` 예제에서는 EBS 루트 볼륨의 지정된 스냅샷을 AMI 사용하여 를 디바이스로 등록합니다/dev/xvda. 블록 디바이스 매핑에는 빈 100GiB EBS 볼륨도 디바이스 로 포함됩니다/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}
```

출력:

```
{
    "ImageId": "ami-1a2b3c4d5eEXAMPLE"
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon Machine Images(AMI)를 참조하세요. EC2

- 자세한 API 내용은 명령 참조RegisterImage의 섹션을 참조하세요. AWS CLI

### register-instance-event-notification-attributes

다음 코드 예시에서는 register-instance-event-notification-attributes을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 이벤트 알림에 모든 태그를 포함하려면

다음 register-instance-event-notification-attributes 예제에서는 이벤트 알림의 모든 태그를 포함합니다.

```
aws ec2 register-instance-event-notification-attributes \
    --instance-tag-attribute IncludeAllTagsOfInstance=true
```

출력:

```
{
    "InstanceTagAttribute": {
        "InstanceTagKeys": [],
        "IncludeAllTagsOfInstance": true
    }
}
```

자세한 내용은 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 인스턴스에 대해 예약된 이벤트를 참조하세요.

예제 2: 이벤트 알림에 특정 태그를 포함하려면

다음 register-instance-event-notification-attributes 예제에서는 이벤트 알림에 지정된 태그를 포함합니다. 이 IncludeAllTagsOfInstance인 경우 태그를 지정할 수 없습니다true.

```
aws ec2 register-instance-event-notification-attributes \
    --instance-tag-attribute InstanceTagKeys="tag-key1","tag-key2"
```

출력:

```
{
    "InstanceTagAttribute": {
        "InstanceTagKeys": [
            "tag-key1",
```

```
            "tag-key2"
        ],
        "IncludeAllTagsOfInstance": false
    }
}
```

자세한 내용은 Linux 인스턴스용 Amazon Elastic Compute Cloud 사용 설명서의 인스턴스에 대해 예약된 이벤트를 참조하세요.

- 자세한 API 내용은 명령 참조RegisterInstanceEventNotificationAttributes의 섹션을 참조하세요. AWS CLI

### register-transit-gateway-multicase-group-sources

다음 코드 예시에서는 register-transit-gateway-multicase-group-sources을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 멀티캐스트 그룹에 소스를 등록하려면

다음 register-transit-gateway-multicast-group-sources 예제에서는 지정된 네트워크 인터페이스 그룹 소스를 멀티캐스트 그룹에 등록합니다.

```
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
```

출력:

```
{
    "RegisteredMulticastGroupSources": {
        "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef79d6e597",
        "RegisteredNetworkInterfaceIds": [
            "eni-07f290fc3c090cbae"
        ],
        "GroupIpAddress": "224.0.1.0"
    }
}
```

자세한 내용은 Transit Gateways 사용 설명서의 멀티캐스트 그룹에 소스 등록을 참조하세요. AWS

- 자세한 API 내용은 명령 참조RegisterTransitGatewayMulticaseGroupSources의 섹션을 참조하세요. AWS CLI

### register-transit-gateway-multicast-group-members

다음 코드 예시에서는 register-transit-gateway-multicast-group-members을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 멀티캐스트 도메인 연결에 대한 정보를 보려면

다음 register-transit-gateway-multicast-group-members 예제에서는 지정된 멀티캐스트 도메인에 대한 연결을 반환합니다.

```
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
```

출력:

```
{
    "RegisteredMulticastGroupMembers": {
        "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef79d6e597",
        "RegisteredNetworkInterfaceIds": [
            "eni-0e246d32695012e81"
        ],
        "GroupIpAddress": "224.0.1.0"
    }
}
```

자세한 내용은 Transit Gateways 사용 설명서의 멀티캐스트 도메인 관리를 참조하세요.

- 자세한 API 내용은 명령 참조RegisterTransitGatewayMulticastGroupMembers의 섹션을 참조하세요. AWS CLI

### register-transit-gateway-multicast-group-sources

다음 코드 예시에서는 register-transit-gateway-multicast-group-sources을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 멀티캐스트 그룹에 소스를 등록하려면

다음 `register-transit-gateway-multicast-group-sources` 예제에서는 지정된 네트워크 인터페이스 그룹 소스를 멀티캐스트 그룹에 등록합니다.

```
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
```

출력:

```
{
    "RegisteredMulticastGroupSources": {
        "TransitGatewayMulticastDomainId": "tgw-mcast-domain-0c4905cef79d6e597",
        "RegisteredNetworkInterfaceIds": [
            "eni-07f290fc3c090cbae"
        ],
        "GroupIpAddress": "224.0.1.0"
    }
}
```

자세한 내용은 Transit Gateways 가이드의 멀티캐스트 도메인 관리를 참조하세요.

- 자세한 API 내용은 명령 참조RegisterTransitGatewayMulticastGroupSources의 섹션을 참조하세요. AWS CLI

### `reject-transit-gateway-peering-attachment`

다음 코드 예시에서는 `reject-transit-gateway-peering-attachment`을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 피어링 연결을 거부하려면

다음 `reject-transit-gateway-peering-attachment` 예제에서는 지정된 전송 게이트웨이 피어링 연결 요청을 거부합니다. `--region` 파라미터는 수락자 전송 게이트웨이가 위치한 리전을 지정합니다.

```
aws ec2 reject-transit-gateway-peering-attachment \
    --transit-gateway-attachment-id tgw-attach-4455667788aabbccd \
    --region us-east-2
```

출력:

```
{
    "TransitGatewayPeeringAttachment": {
        "TransitGatewayAttachmentId": "tgw-attach-4455667788aabbccd",
        "RequesterTgwInfo": {
            "TransitGatewayId": "tgw-123abc05e04123abc",
            "OwnerId": "123456789012",
            "Region": "us-west-2"
        },
        "AccepterTgwInfo": {
            "TransitGatewayId": "tgw-11223344aabbcc112",
            "OwnerId": "123456789012",
            "Region": "us-east-2"
        },
        "State": "rejecting",
        "CreationTime": "2019-12-09T11:50:31.000Z"
    }
}
```

자세한 내용은 Transit Gateways 가이드의 Transit Gateway 피어링 연결을 참조하세요.

- 자세한 API 내용은 명령 참조RejectTransitGatewayPeeringAttachment의 섹션을 참조하세요. AWS CLI

### `reject-transit-gateway-vpc-attachment`

다음 코드 예시에서는 `reject-transit-gateway-vpc-attachment`을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 VPC 연결을 거부하려면

다음 `reject-transit-gateway-vpc-attachment` 예제에서는 지정된 전송 게이트웨이 VPC 연결을 거부합니다.

```
aws ec2 reject-transit-gateway-vpc-attachment \
```

```
    --transit-gateway-attachment-id tgw-attach-0a34fe6b4fEXAMPLE
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 Transit Gateways Guide의 Transit Gateway attachments to VPC a를 참조하세요.

- 자세한 API 내용은 명령 참조RejectTransitGatewayVpcAttachment의 섹션을 참조하세요. AWS CLI

### reject-transit-gateway-vpc-attachments

다음 코드 예시에서는 reject-transit-gateway-vpc-attachments을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 VPC 연결을 거부하려면

다음 reject-transit-gateway-vpc-attachment 예제에서는 지정된 전송 게이트웨이 VPC 연결을 거부합니다.

```
aws ec2 reject-transit-gateway-vpc-attachment \
    --transit-gateway-attachment-id tgw-attach-0a34fe6b4fEXAMPLE
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 Transit Gateways Guide의 Transit Gateway attachments to VPC a를 참조하세요.

- 자세한 API 내용은 명령 참조RejectTransitGatewayVpcAttachments의 섹션을 참조하세요. AWS CLI

### `reject-vpc-endpoint-connections`

다음 코드 예시에서는 `reject-vpc-endpoint-connections`을 사용하는 방법을 보여 줍니다.

AWS CLI

인터페이스 엔드포인트 연결 요청을 거부하려면

이 예제에서는 지정된 엔드포인트 서비스에 대한 지정된 엔드포인트 연결 요청을 거부합니다.

명령:

```
aws ec2 reject-vpc-endpoint-connections --service-id vpce-svc-03d5ebb7d9579a2b3 --
vpc-endpoint-ids vpce-0c1308d7312217abc
```

출력:

```
{
```

```
  "Unsuccessful": []
}
```

- 자세한 API 내용은 명령 참조RejectVpcEndpointConnections의 섹션을 참조하세요. AWS CLI

### `reject-vpc-peering-connection`

다음 코드 예시에서는 `reject-vpc-peering-connection`을 사용하는 방법을 보여 줍니다.

AWS CLI

VPC 피어링 연결을 거부하려면

이 예제에서는 지정된 VPC 피어링 연결 요청을 거부합니다.

명령:

```
aws ec2 reject-vpc-peering-connection --vpc-peering-connection-id pcx-1a2b3c4d
```

출력:

```
{
    "Return": true
}
```

- 자세한 API 내용은 명령 참조RejectVpcPeeringConnection의 섹션을 참조하세요. AWS CLI

### `release-address`

다음 코드 예시에서는 `release-address`을 사용하는 방법을 보여 줍니다.

AWS CLI

EC2-Classic에 대한 탄력적 IP 주소를 릴리스하려면

이 예제에서는 EC2-Classic의 인스턴스에 사용할 탄력적 IP 주소를 릴리스합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 release-address --public-ip 198.51.100.0
```

EC2에 대한 탄력적 IP 주소를 해제하려면VPC

이 예제에서는 의 인스턴스와 함께 사용할 탄력적 IP 주소를 릴리스합니다VPC. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 release-address --allocation-id eipalloc-64d5890a
```

- 자세한 API 내용은 명령 참조ReleaseAddress의 섹션을 참조하세요. AWS CLI

### release-hosts

다음 코드 예시에서는 release-hosts을 사용하는 방법을 보여 줍니다.

AWS CLI

계정에서 전용 호스트를 해제하려면

계정에서 전용 호스트를 해제합니다. 호스트에 있는 인스턴스는 호스트를 릴리스하기 전에 중지하거나 종료해야 합니다.

명령:

```
aws ec2 release-hosts --host-id=h-0029d6e3cacf1b3da
```

출력:

```
{
    "Successful":  [
        "h-0029d6e3cacf1b3da"
         ],
  "Unsuccessful": []

 }
```

- 자세한 API 내용은 명령 참조ReleaseHosts의 섹션을 참조하세요. AWS CLI

### release-ipam-pool-allocation

다음 코드 예시에서는 release-ipam-pool-allocation을 사용하는 방법을 보여 줍니다.

AWS CLI

IPAM 풀 할당을 릴리스하려면

이 예제에서는 풀을 삭제하려고 했지만 IPAM 풀에 할당이 있는 동안에는 풀을 삭제할 수 없다는 오류가 발생한 IPAM 위임된 관리자입니다. 이 명령을 사용하여 풀 할당을 해제합니다.

유의할 사항:

이 명령은 사용자 지정 할당에만 사용할 수 있습니다. 리소스를 삭제하지 않고 리소스에 대한 할당을 제거하려면 를 사용하여 모니터링된 상태를 false로 설정합니다modify-ipam-resource-cidr. 이 요청을 완료하려면 IPAM 풀 ID가 필요하며, 이 ID는 로 가져올 수 있습니다describe-ipam-pools. 또한 할당 ID가 필요합니다get-ipam-pool-allocations. 를 사용하여 할당을 하나씩 제거하지 않으려면 IPAM 풀을 삭제할 `--cascade option` 때 를 사용하여 풀에서 할당을 삭제하기 전에 자동으로 해제할 수 있습니다. 이 명령을 실행하기 전에 여러 사전 조건이 있습니다. 자세한 내용은 Amazon VPC IPAM 사용 설명서의 할당 해제를 참조하세요. 이 명령을 실행하는 `--region` 는 할당이 있는 IPAM 풀의 로캘이어야 합니다.

다음 `release-ipam-pool-allocation` 예제에서는 IPAM 풀 할당을 릴리스합니다.

```
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
```

출력:

```
{
    "Success": true
}
```

할당을 해제하면 를 실행할 수 있습니다delete-ipam-pool.

- 자세한 API 내용은 명령 참조ReleaseIpamPoolAllocation의 섹션을 참조하세요. AWS CLI

### replace-iam-instance-profile-association

다음 코드 예시에서는 `replace-iam-instance-profile-association`을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 인스턴스의 인스턴스 프로파일을 교체하려면

이 예제는 연결로 표시되는 IAM 인스턴스 프로파일을 라는 IAM 인스턴스 프로파일iip-assoc-060bae234aac2e7fa로 바꿉니다AdminRole.

```
aws ec2 replace-iam-instance-profile-association \
    --iam-instance-profile Name=AdminRole \
    --association-id iip-assoc-060bae234aac2e7fa
```

출력:

```
{
    "IamInstanceProfileAssociation": {
        "InstanceId": "i-087711ddaf98f9489",
        "State": "associating",
        "AssociationId": "iip-assoc-0b215292fab192820",
        "IamInstanceProfile": {
            "Id": "AIPAJLNLDX3AMYZNWYYAY",
            "Arn": "arn:aws:iam::123456789012:instance-profile/AdminRole"
        }
    }
}
```

- 자세한 API 내용은 명령 참조ReplaceIamInstanceProfileAssociation의 섹션을 참조하세요. AWS CLI

### replace-network-acl-association

다음 코드 예시에서는 replace-network-acl-association을 사용하는 방법을 보여 줍니다.

AWS CLI

서브넷과 ACL 연결된 네트워크를 교체하려면

이 예제에서는 지정된 네트워크를 지정된 네트워크 ACL 연결ACL의 서브넷과 연결합니다.

명령:

```
aws ec2 replace-network-acl-association --association-id aclassoc-e5b95c8c --
network-acl-id acl-5fb85d36
```

출력:

```
{
    "NewAssociationId": "aclassoc-3999875b"
}
```

- 자세한 API 내용은 명령 참조ReplaceNetworkAclAssociation의 섹션을 참조하세요. AWS CLI

### replace-network-acl-entry

다음 코드 예시에서는 replace-network-acl-entry을 사용하는 방법을 보여 줍니다.

AWS CLI

네트워크 ACL 항목을 바꾸려면

이 예제는 지정된 네트워크 에 대한 항목을 대체합니다ACL. 새 규칙 100은 UDP 포트 53(DNS)의 203.0.113.12/24에서 연결된 서브넷으로 트래픽을 수신할 수 있습니다.

명령:

```
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
```

- 자세한 API 내용은 명령 참조ReplaceNetworkAclEntry의 섹션을 참조하세요. AWS CLI

### replace-route-table-association

다음 코드 예시에서는 replace-route-table-association을 사용하는 방법을 보여 줍니다.

AWS CLI

서브넷과 연결된 라우팅 테이블을 교체하려면

이 예제에서는 지정된 라우팅 테이블을 지정된 라우팅 테이블 연결의 서브넷과 연결합니다.

명령:

```
aws ec2 replace-route-table-association --association-id rtbassoc-781d0d1a --route-table-id rtb-22574640
```

출력:

```
{
    "NewAssociationId": "rtbassoc-3a1f0f58"
}
```

- 자세한 API 내용은 명령 참조ReplaceRouteTableAssociation의 섹션을 참조하세요. AWS CLI

### replace-route

다음 코드 예시에서는 replace-route을 사용하는 방법을 보여 줍니다.

AWS CLI

경로를 바꾸려면

이 예제는 지정된 라우팅 테이블의 지정된 라우팅을 대체합니다. 새 경로는 지정된 경로와 일치 CIDR하고 트래픽을 지정된 가상 프라이빗 게이트웨이로 전송합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 replace-route --route-table-id rtb-22574640 --destination-cidr-
block 10.0.0.0/16 --gateway-id vgw-9a4cacf3
```

- 자세한 API 내용은 명령 참조ReplaceRoute의 섹션을 참조하세요. AWS CLI

### replace-transit-gateway-route

다음 코드 예시에서는 replace-transit-gateway-route을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 전송 게이트웨이 라우팅 테이블에서 지정된 라우팅을 바꾸려면

다음 replace-transit-gateway-route 예제는 지정된 전송 게이트웨이 라우팅 테이블의 라우팅을 대체합니다.

```
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
```

출력:

```
{
    "Route": {
        "DestinationCidrBlock": "10.0.2.0/24",
        "TransitGatewayAttachments": [
            {
                "ResourceId": "vpc-4EXAMPLE",
                "TransitGatewayAttachmentId": "tgw-attach-09b52ccdb5EXAMPLE",
                "ResourceType": "vpc"
            }
        ],
        "Type": "static",
        "State": "active"
    }
}
```

자세한 내용은 Transit Gateways 가이드의 Transit Gateway route table을 참조하세요.

- 자세한 API 내용은 명령 참조ReplaceTransitGatewayRoute의 섹션을 참조하세요. AWS CLI

### report-instance-status

다음 코드 예시에서는 report-instance-status을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스에 대한 상태 피드백을 보고하려면

이 예제 명령은 지정된 인스턴스에 대한 상태 피드백을 보고합니다.

명령:

```
aws ec2 report-instance-status --instances i-1234567890abcdef0 --status impaired --
reason-codes unresponsive
```

- 자세한 API 내용은 명령 참조ReportInstanceStatus의 섹션을 참조하세요. AWS CLI

### request-spot-fleet

다음 코드 예시에서는 request-spot-fleet을 사용하는 방법을 보여 줍니다.

AWS CLI

서브넷에서 가장 저렴한 가격으로 스팟 플릿을 요청하려면

이 예제 명령은 서브넷마다 다른 두 가지 시작 사양으로 스팟 플릿 요청을 생성합니다. 스팟 플릿은 지정된 서브넷에서 최저 가격으로 인스턴스를 시작합니다. 인스턴스가 기본 에서 시작되면 기본적으로 퍼블릭 IP 주소를 수신VPC합니다. 인스턴스가 기본값이 아닌 에서 시작VPC되는 경우 기본적으로 퍼블릭 IP 주소를 수신하지 않습니다.

스팟 플릿 요청에서 동일한 가용 영역과 다른 서브넷을 지정할 수 없습니다.

명령:

```
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"
          }
      }
  ]
}
```

출력:

```
{
  "SpotFleetRequestId": "sfr-73fbd2ce-aa30-494c-8788-1cee4EXAMPLE"
}
```

가용 영역에서 최저 가격으로 스팟 플릿을 요청하려면

이 예제 명령은 가용 영역에 따라 다른 두 가지 시작 사양으로 스팟 플릿 요청을 생성합니다. 스팟 플릿은 지정된 가용 영역에서 최저 가격으로 인스턴스를 시작합니다. 계정이 EC2만VPC 지원하는 경우 Amazon은 가용 영역의 기본 서브넷에서 스팟 인스턴스를 EC2 시작합니다. 계정이 EC2-Classic을 지원하는 경우 Amazon은 가용 영역의 EC2-Classic에서 인스턴스를 EC2 시작합니다.

명령:

```
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"
          }
      }
  ]
}
```

서브넷에서 스팟 인스턴스를 시작하고 퍼블릭 IP 주소를 할당하려면

이 예제 명령은 기본값이 아닌 에서 시작된 인스턴스에 퍼블릭 주소를 할당합니다VPC. 네트워크 인터페이스를 지정할 때는 네트워크 인터페이스를 사용하여 서브넷 ID와 보안 그룹 ID를 포함해야 합니다.

명령:

```
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"
          }
      }
  ]
}
```

분산 할당 전략을 사용하여 스팟 플릿을 요청하려면

이 예제 명령은 다양한 할당 전략을 사용하여 30개의 인스턴스를 시작하는 스팟 플릿 요청을 생성합니다. 시작 사양은 인스턴스 유형에 따라 다릅니다. 스팟 플릿은 각 유형의 인스턴스가 10개 있도록 시작 사양에 걸쳐 인스턴스를 배포합니다.

명령:

```
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"
      }
  ]
}
```

자세한 내용은 Amazon Elastic Compute Cloud 사용 설명서의 스팟 플릿 요청을 참조하세요.

- 자세한 API 내용은 명령 참조RequestSpotFleet의 섹션을 참조하세요. AWS CLI

### `request-spot-instances`

다음 코드 예시에서는 `request-spot-instances`을 사용하는 방법을 보여 줍니다.

AWS CLI

스팟 인스턴스를 요청하려면

이 예제 명령은 지정된 가용 영역의 인스턴스 5개에 대한 일회성 스팟 인스턴스 요청을 생성합니다. 계정이 EC2만VPC 지원하는 경우 Amazon은 지정된 가용 영역의 기본 서브넷에서 인스턴스

를 EC2 시작합니다. 계정이 EC2-Classic을 지원하는 경우 Amazon은 지정된 가용 영역의 EC2-Classic에서 인스턴스를 EC2 시작합니다.

명령:

```
aws ec2 request-spot-instances --spot-price "0.03" --instance-count 5 --type "one-time" --launch-specification file://specification.json
```

Specification.json:

```
{
  "ImageId": "ami-1a2b3c4d",
  "KeyName": "my-key-pair",
  "SecurityGroupIds": [ "sg-1a2b3c4d" ],
  "InstanceType": "m3.medium",
  "Placement": {
    "AvailabilityZone": "us-west-2a"
  },
  "IamInstanceProfile": {
      "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
  }
}
```

출력:

```
{
  "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"
        },
        ...
    ]
}
```

이 예제 명령은 지정된 서브넷의 인스턴스 5개에 대한 일회성 스팟 인스턴스 요청을 생성합니다. Amazon은 지정된 서브넷에서 인스턴스를 EC2 시작합니다. VPC 가 기본값이 아닌 VPC인 경우 인스턴스는 기본적으로 퍼블릭 IP 주소를 수신하지 않습니다.

명령:

```
aws ec2 request-spot-instances --spot-price "0.050" --instance-count 5 --type "one-time" --launch-specification file://specification.json
```

Specification.json:

```
{
  "ImageId": "ami-1a2b3c4d",
  "SecurityGroupIds": [ "sg-1a2b3c4d" ],
  "InstanceType": "m3.medium",
  "SubnetId": "subnet-1a2b3c4d",
  "IamInstanceProfile": {
      "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
  }
```

```
}
```

출력:

```
{
  "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"
      },
      ...
  ]
}
```

이 예제에서는 기본값이 아닌 에서 시작하는 스팟 인스턴스에 퍼블릭 IP 주소를 할당합니다VPC. 네트워크 인터페이스를 지정할 때는 네트워크 인터페이스를 사용하여 서브넷 ID와 보안 그룹 ID를 포함해야 합니다.

명령:

```
aws ec2 request-spot-instances --spot-price "0.050" --instance-count 1 --type "one-time" --launch-specification file://specification.json
```

Specification.json:

```
{
  "ImageId": "ami-1a2b3c4d",
  "KeyName": "my-key-pair",
  "InstanceType": "m3.medium",
  "NetworkInterfaces": [
    {
      "DeviceIndex": 0,
      "SubnetId": "subnet-1a2b3c4d",
      "Groups": [ "sg-1a2b3c4d" ],
      "AssociatePublicIpAddress": true
    }
  ],
  "IamInstanceProfile": {
      "Arn": "arn:aws:iam::123456789012:instance-profile/my-iam-role"
  }
}
```

- 자세한 API 내용은 명령 참조RequestSpotInstances의 섹션을 참조하세요. AWS CLI

### `reset-address-attribute`

다음 코드 예시에서는 `reset-address-attribute`을 사용하는 방법을 보여 줍니다.

AWS CLI

탄력적 IP 주소와 연결된 도메인 이름 속성을 재설정하려면

다음 `reset-address-attribute` 예제에서는 탄력적 IP 주소의 도메인 이름 속성을 재설정합니다.

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
```

출력:

```
{
    "Addresses": [
        {
            "PublicIp": "192.0.2.0",
            "AllocationId": "eipalloc-abcdef01234567890",
            "PtrRecord": "example.com."
            "PtrRecordUpdate": {
                "Value": "example.net.",
                "Status": "PENDING"
        }
    ]
}
```

보류 중인 변경 사항을 모니터링하려면 명령 참조describe-addresses-attribute의 섹션을 참조하세요. AWS CLI

- 자세한 API 내용은 명령 참조ResetAddressAttribute의 섹션을 참조하세요. AWS CLI

### `reset-ebs-default-kms-key-id`

다음 코드 예시에서는 `reset-ebs-default-kms-key-id`을 사용하는 방법을 보여 줍니다.

AWS CLI

EBS 암호화 CMK 기본값을 재설정하려면

다음 `reset-ebs-default-kms-key-id` 예제에서는 현재 리전의 AWS 계정에 CMK 대한 EBS 암호화 기본값을 재설정합니다.

```
aws ec2 reset-ebs-default-kms-key-id
```

출력:

```
{
   "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/8c5b2c63-b9bc-45a3-
a87a-5513eEXAMPLE"
}
```

- 자세한 API 내용은 명령 참조ResetEbsDefaultKmsKeyId의 섹션을 참조하세요. AWS CLI

### reset-fpga-image-attribute

다음 코드 예시에서는 reset-fpga-image-attribute을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon FPGA 이미지의 속성을 재설정하려면

이 예제에서는 지정된 에 대한 로드 권한을 재설정합니다AFI.

명령:

```
aws ec2 reset-fpga-image-attribute --fpga-image-id afi-0d123e123bfc85abc --
attribute loadPermission
```

출력:

```
{
  "Return": true
}
```

- 자세한 API 내용은 명령 참조ResetFpgaImageAttribute의 섹션을 참조하세요. AWS CLI

### reset-image-attribute

다음 코드 예시에서는 reset-image-attribute을 사용하는 방법을 보여 줍니다.

AWS CLI

launchPermission 속성을 재설정하려면

이 예제에서는 지정된 의 launchPermission 속성을 기본값AMI으로 재설정합니다. 기본적으로 AMIs는 프라이빗입니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 reset-image-attribute --image-id ami-5731123e --attribute launchPermission
```

- 자세한 API 내용은 명령 참조ResetImageAttribute의 섹션을 참조하세요. AWS CLI

### reset-instance-attribute

다음 코드 예시에서는 reset-instance-attribute을 사용하는 방법을 보여 줍니다.

AWS CLI

sourceDestCheck 속성을 재설정하려면

이 예제에서는 지정된 인스턴스의 sourceDestCheck 속성을 재설정합니다. 인스턴스는 에 있어야 합니다VPC. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 reset-instance-attribute --instance-id i-1234567890abcdef0 --
attribute sourceDestCheck
```

커널 속성을 재설정하려면

이 예제에서는 지정된 인스턴스의 kernel 속성을 재설정합니다. 인스턴스는 stopped 상태여야 합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 reset-instance-attribute --instance-id i-1234567890abcdef0 --
attribute kernel
```

ramdisk 속성을 재설정하려면

이 예제에서는 지정된 인스턴스의 ramdisk 속성을 재설정합니다. 인스턴스는 stopped 상태여야 합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 reset-instance-attribute --instance-id i-1234567890abcdef0 --
attribute ramdisk
```

- 자세한 API 내용은 명령 참조ResetInstanceAttribute의 섹션을 참조하세요. AWS CLI

### reset-network-interface-attribute

다음 코드 예시에서는 reset-network-interface-attribute을 사용하는 방법을 보여 줍니다.

AWS CLI

네트워크 인터페이스 속성을 재설정하려면

다음 reset-network-interface-attribute 예제에서는 소스/대상 검사 속성의 값을 로 재설정합니다true.

```
aws ec2 reset-network-interface-attribute \
    --network-interface-id eni-686ea200 \
    --source-dest-check
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조ResetNetworkInterfaceAttribute의 섹션을 참조하세요. AWS CLI

### reset-snapshot-attribute

다음 코드 예시에서는 reset-snapshot-attribute을 사용하는 방법을 보여 줍니다.

AWS CLI

스냅샷 속성을 재설정하려면

이 예제에서는 스냅샷 에 대한 볼륨 생성 권한을 재설정합니다snap-1234567890abcdef0. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 reset-snapshot-attribute --snapshot-id snap-1234567890abcdef0 --
attribute createVolumePermission
```

- 자세한 API 내용은 명령 참조ResetSnapshotAttribute의 섹션을 참조하세요. AWS CLI

### restore-address-to-classic

다음 코드 예시에서는 restore-address-to-classic을 사용하는 방법을 보여 줍니다.

AWS CLI

주소를 EC2-Classic으로 복원하려면

이 예제는 Elastic IP 주소 198.51.100.0을 EC2-Classic 플랫폼으로 복원합니다.

명령:

```
aws ec2 restore-address-to-classic --public-ip 198.51.100.0
```

출력:

```
{
  "Status": "MoveInProgress",
  "PublicIp": "198.51.100.0"
}
```

- 자세한 API 내용은 명령 참조RestoreAddressToClassic의 섹션을 참조하세요. AWS CLI

### restore-image-from-recycle-bin

다음 코드 예시에서는 restore-image-from-recycle-bin을 사용하는 방법을 보여 줍니다.

AWS CLI

휴지통에서 이미지를 복원하려면

다음 restore-image-from-recycle-bin 예제에서는 휴지통에서 AMI ami-0111222333444abcd를 복원합니다.

```
aws ec2 restore-image-from-recycle-bin \
    --image-id ami-0111222333444abcd
```

출력:

```
{
    "Return": true
```

```
}
```

자세한 내용은 Amazon Elastic Compute Cloud 사용 설명서의 휴지통AMIs에서 복구를 참조하세요.

- 자세한 API 내용은 명령 참조RestoreImageFromRecycleBin의 섹션을 참조하세요. AWS CLI

### `restore-managed-prefix-list-version`

다음 코드 예시에서는 `restore-managed-prefix-list-version`을 사용하는 방법을 보여 줍니다.

AWS CLI

us-west-2**접두사 목록 버전을 복원하려면**

다음은 지정된 접두사 목록의 버전 1에서 항목을 `restore-managed-prefix-list-version` 복원합니다.

```
aws ec2 restore-managed-prefix-list-version \
    --prefix-list-id pl-0123456abcabcabc1 \
    --current-version 2 \
    --previous-version 1
```

출력:

```
{
    "PrefixList": {
        "PrefixListId": "pl-0123456abcabcabc1",
        "AddressFamily": "IPv4",
        "State": "restore-in-progress",
        "PrefixListArn": "arn:aws:ec2:us-west-2:123456789012:prefix-list/
pl-0123456abcabcabc1",
        "PrefixListName": "vpc-cidrs",
        "MaxEntries": 10,
        "Version": 2,
        "OwnerId": "123456789012"
    }
}
```

자세한 내용은 Amazon VPC 사용 설명서의 관리형 접두사 목록을 참조하세요.

- 자세한 API 내용은 명령 참조RestoreManagedPrefixListVersion의 섹션을 참조하세요. AWS CLI

### restore-snapshot-from-recycle-bin

다음 코드 예시에서는 restore-snapshot-from-recycle-bin을 사용하는 방법을 보여 줍니다.

AWS CLI

휴지통에서 스냅샷을 복원하려면

다음 restore-snapshot-from-recycle-bin 예제에서는 휴지통에서 스냅샷을 복원합니다. 휴지통에서 스냅샷을 복원하면 스냅샷을 즉시 사용할 수 있으며 휴지통에서 스냅샷이 제거됩니다. 계정의 다른 스냅샷을 사용하는 것과 동일한 방식으로 복원된 스냅샷을 사용할 수 있습니다.

```
aws ec2 restore-snapshot-from-recycle-bin \
    --snapshot-id snap-01234567890abcdef
```

이 명령은 출력을 생성하지 않습니다.

Amazon용 휴지통에 대한 자세한 내용은 Amazon EC2 사용 설명서의 휴지통에서 스냅샷 복구를 EBS참조하세요.

- 자세한 API 내용은 명령 참조RestoreSnapshotFromRecycleBin의 섹션을 참조하세요. AWS CLI

### restore-snapshot-tier

다음 코드 예시에서는 restore-snapshot-tier을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 아카이브된 스냅샷을 영구적으로 복원하려면

다음 restore-snapshot-tier 예제에서는 지정된 스냅샷을 영구적으로 복원합니다. 를 지정--snapshot-id하고 permanent-restore 옵션을 포함합니다.

```
aws ec2 restore-snapshot-tier \
    --snapshot-id snap-01234567890abcedf \
    --permanent-restore
```

출력:

```
{
    "SnapshotId": "snap-01234567890abcedf",
    "IsPermanentRestore": true
```

```
}
```

스냅샷 아카이빙에 대한 자세한 내용은 Amazon  EC2 사용 설명서의 Archive Amazon EBS 스냅샷 <https://docs.aws.amazon.com/AWS EC2/latest/UserGuide/snapshot-archive.html>을 참조하세요.

예제 2: 보관된 스냅샷을 일시적으로 복원하려면

다음 `restore-snapshot-tier` 예제에서는 지정된 스냅샷을 일시적으로 복원합니다. `--permanent-restore` 옵션을 생략합니다. 에 `--snapshot-id` 및 를 지정하고 스냅샷을 복원할 일수를 `temporary-restore-days`지정합니다. 는 일수로 지정`temporary-restore-days`해야 합니다. 허용되는 범위는 1 ~입니다`180`. 값을 지정하지 않으면 기본적으로 1일이 사용됩니다.

```
aws ec2 restore-snapshot-tier \
    --snapshot-id snap-01234567890abcedf \
    --temporary-restore-days 5
```

출력:

```
{
    "SnapshotId": "snap-01234567890abcedf",
    "RestoreDuration": 5,
    "IsPermanentRestore": false
}
```

스냅샷 아카이빙에 대한 자세한 내용은 Amazon  EC2 사용 설명서의 Archive Amazon EBS 스냅샷 <https://docs.aws.amazon.com/AWS EC2/latest/UserGuide/snapshot-archive.html>을 참조하세요.

예제 3: 복원 기간을 수정하려면

다음 `restore-snapshot-tier` 예제에서는 지정된 스냅샷의 복원 기간을 `10` 일 단위로 변경합니다.

```
aws ec2 restore-snapshot-tier \
    --snapshot-id snap-01234567890abcedf
    --temporary-restore-days 10
```

출력:

```
{
    "SnapshotId": "snap-01234567890abcedf",
    "RestoreDuration": 10,
```

```
    "IsPermanentRestore": false
}
```

스냅샷 아카이빙에 대한 자세한 내용은 Amazon  EC2 사용 설명서의 Archive Amazon EBS 스냅샷 <https://docs.aws.amazon.com/AWS EC2/latest/UserGuide/snapshot-archive.html>을 참조하세요.

예제 4: 복원 유형 수정

다음 `restore-snapshot-tier` 예제에서는 지정된 스냅샷의 복원 유형을 임시에서 영구으로 변경합니다.

```
aws ec2 restore-snapshot-tier \
    --snapshot-id snap-01234567890abcedf
    --permanent-restore
```

출력:

```
{
    "SnapshotId": "snap-01234567890abcedf",
    "IsPermanentRestore": true
}
```

스냅샷 아카이빙에 대한 자세한 내용은 Amazon  EC2 사용 설명서의 Archive Amazon EBS 스냅샷 <https://docs.aws.amazon.com/AWS EC2/latest/UserGuide/snapshot-archive.html>을 참조하세요.

- 자세한 API 내용은 명령 참조RestoreSnapshotTier의 섹션을 참조하세요. AWS CLI

### `revoke-client-vpn-ingress`

다음 코드 예시에서는 `revoke-client-vpn-ingress`을 사용하는 방법을 보여 줍니다.

AWS CLI

클라이언트 VPN 엔드포인트에 대한 권한 부여 규칙을 취소하려면

다음 `revoke-client-vpn-ingress` 예제에서는 모든 그룹에 대한 인터넷 액세스 규칙 (`0.0.0.0/0`)을 취소합니다.

```
aws ec2 revoke-client-vpn-ingress \
    --client-vpn-endpoint-id cvpn-endpoint-123456789123abcde \
    --target-network-cidr 0.0.0.0/0 --revoke-all-groups
```

출력:

```
{
    "Status": {
        "Code": "revoking"
    }
}
```

자세한 내용은 AWS 클라이언트 VPN 관리자 안내서의 권한 부여 규칙을 참조하세요.

- 자세한 API 내용은 명령 참조RevokeClientVpnIngress의 섹션을 참조하세요. AWS CLI

### `revoke-security-group-egress`

다음 코드 예시에서는 `revoke-security-group-egress`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 특정 주소 범위로 아웃바운드 트래픽을 허용하는 규칙을 제거하려면

다음 `revoke-security-group-egress` 예제 명령은 TCP 포트 80에서 지정된 주소 범위에 대한 액세스 권한을 부여하는 규칙을 제거합니다.

```
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}]
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon EC2 사용 설명서의 보안 그룹을 참조하세요.

예제 2: 특정 보안 그룹에 대한 아웃바운드 트래픽을 허용하는 규칙을 제거하려면

다음 `revoke-security-group-egress` 예제 명령은 TCP 포트 80에서 지정된 보안 그룹에 대한 액세스 권한을 부여하는 규칙을 제거합니다.

```
aws ec2 revoke-security-group-egress \
    --group-id sg-026c12253ce15eff7 \
    --ip-permissions '[{"IpProtocol": "tcp", "FromPort": 443, "ToPort":
 443,"UserIdGroupPairs": [{"GroupId": "sg-06df23a01ff2df86d"}]}]'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon EC2 사용 설명서의 보안 그룹을 참조하세요.

- 자세한 API 내용은 명령 참조RevokeSecurityGroupEgress의 섹션을 참조하세요. AWS CLI

### `revoke-security-group-ingress`

다음 코드 예시에서는 `revoke-security-group-ingress`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 보안 그룹에서 규칙을 제거하려면

다음 `revoke-security-group-ingress` 예제에서는 기본 에 지정된 보안 그룹에서 `203.0.113.0/24` 주소 범위에 대한 TCP 포트 22 액세스를 제거합니다VPC.

```
aws ec2 revoke-security-group-ingress \
    --group-name mySecurityGroup
    --protocol tcp \
    --port 22 \
    --cidr 203.0.113.0/24
```

이 명령은 성공하면 출력을 생성하지 않습니다.

자세한 내용은 Amazon EC2 사용 설명서의 보안 그룹을 참조하세요.

예제 2: IP 권한 세트를 사용하여 규칙을 제거하려면

다음 `revoke-security-group-ingress` 예제에서는 `ip-permissions` 파라미터를 사용하여 ICMP 메시지를 허용하는 인바운드 규칙을 제거합니다`Destination Unreachable: Fragmentation Needed and Don't Fragment was Set`(유형 3, 코드 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}]
```

이 명령은 성공하면 출력을 생성하지 않습니다.

자세한 내용은 Amazon EC2 사용 설명서의 보안 그룹을 참조하세요.

- 자세한 API 내용은 명령 참조RevokeSecurityGroupIngress의 섹션을 참조하세요. AWS CLI

### run-instances

다음 코드 예시에서는 run-instances을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 기본 서브넷에서 인스턴스를 시작하는 방법

다음 run-instances 예제에서는 현재 리전의 기본 서브넷t2.micro에 단일 유형의 인스턴스를 시작하고 이를 리전의 기본 서브넷VPC에 연결합니다. (Linux) 또는 SSH (RDPWindows)를 사용하여 인스턴스에 연결할 계획이 없는 경우 키 페어는 선택 사항입니다.

```
aws ec2 run-instances \
    --image-id ami-0abcdef1234567890 \
    --instance-type t2.micro \
    --key-name MyKeyPair
```

출력:

```
{
    "Instances": [
        {
            "AmiLaunchIndex": 0,
            "ImageId": "ami-0abcdef1234567890",
            "InstanceId": "i-1231231230abcdef0",
            "InstanceType": "t2.micro",
            "KeyName": "MyKeyPair",
            "LaunchTime": "2018-05-10T08:05:20.000Z",
            "Monitoring": {
                "State": "disabled"
            },
            "Placement": {
                "AvailabilityZone": "us-east-2a",
                "GroupName": "",
                "Tenancy": "default"
            },
            "PrivateDnsName": "ip-10-0-0-157.us-east-2.compute.internal",
            "PrivateIpAddress": "10.0.0.157",
            "ProductCodes": [],
            "PublicDnsName": "",
            "State": {
                "Code": 0,
                "Name": "pending"
```

```
            },
            "StateTransitionReason": "",
            "SubnetId": "subnet-04a636d18e83cfacb",
            "VpcId": "vpc-1234567890abcdef0",
            "Architecture": "x86_64",
            "BlockDeviceMappings": [],
            "ClientToken": "",
            "EbsOptimized": false,
            "Hypervisor": "xen",
            "NetworkInterfaces": [
                {
                    "Attachment": {
                        "AttachTime": "2018-05-10T08:05:20.000Z",
                        "AttachmentId": "eni-attach-0e325c07e928a0405",
                        "DeleteOnTermination": true,
                        "DeviceIndex": 0,
                        "Status": "attaching"
                    },
                    "Description": "",
                    "Groups": [
                        {
                            "GroupName": "MySecurityGroup",
                            "GroupId": "sg-0598c7d356eba48d7"
                        }
                    ],
                    "Ipv6Addresses": [],
                    "MacAddress": "0a:ab:58:e0:67:e2",
                    "NetworkInterfaceId": "eni-0c0a29997760baee7",
                    "OwnerId": "123456789012",
                    "PrivateDnsName": "ip-10-0-0-157.us-east-2.compute.internal",
                    "PrivateIpAddress": "10.0.0.157",
                    "PrivateIpAddresses": [
                        {
                            "Primary": true,
                            "PrivateDnsName": "ip-10-0-0-157.us-
east-2.compute.internal",
                            "PrivateIpAddress": "10.0.0.157"
                        }
                    ],
                    "SourceDestCheck": true,
                    "Status": "in-use",
                    "SubnetId": "subnet-04a636d18e83cfacb",
                    "VpcId": "vpc-1234567890abcdef0",
                    "InterfaceType": "interface"
```

```
                }
            ],
            "RootDeviceName": "/dev/xvda",
            "RootDeviceType": "ebs",
            "SecurityGroups": [
                {
                    "GroupName": "MySecurityGroup",
                    "GroupId": "sg-0598c7d356eba48d7"
                }
            ],
            "SourceDestCheck": true,
            "StateReason": {
                "Code": "pending",
                "Message": "pending"
            },
            "Tags": [],
            "VirtualizationType": "hvm",
            "CpuOptions": {
                "CoreCount": 1,
                "ThreadsPerCore": 1
            },
            "CapacityReservationSpecification": {
                "CapacityReservationPreference": "open"
            },
            "MetadataOptions": {
                "State": "pending",
                "HttpTokens": "optional",
                "HttpPutResponseHopLimit": 1,
                "HttpEndpoint": "enabled"
            }
        }
    ],
    "OwnerId": "123456789012",
    "ReservationId": "r-02a3f596d91211712"
}
```

예제 2: 기본이 아닌 서브넷에서 인스턴스를 시작하고 퍼블릭 IP 주소를 추가하는 방법

다음 `run-instances` 예제에서는 기본이 아닌 서브넷에서 시작하는 인스턴스에 대해 퍼블릭 IP 주소를 요청합니다. 인스턴스는 지정된 보안 그룹에 연결됩니다.

```
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
```

run-instances 출력 예제는 예제 1을 참조하세요.

예제 3: 추가 볼륨이 포함된 인스턴스를 시작하는 방법

다음 run-instances 예제에서는 시작할 때 추가 볼륨을 연결하도록 mapping.json에 지정된 블록 디바이스 매핑을 사용합니다. 블록 디바이스 매핑은 EBS 볼륨, 인스턴스 스토어 볼륨 또는 EBS 볼륨과 인스턴스 스토어 볼륨을 모두 지정할 수 있습니다.

```
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
```

mapping.json의 콘텐츠. 이 예제/dev/sdh에서는 크기가 100GiB 인 빈 EBS 볼륨을 추가합니다.

```
[
    {
        "DeviceName": "/dev/sdh",
        "Ebs": {
            "VolumeSize": 100
        }
    }
]
```

mapping.json의 콘텐츠. 이 예제에서는 ephemeral1을 인스턴스 스토어 볼륨으로 추가합니다.

```
[
    {
        "DeviceName": "/dev/sdc",
        "VirtualName": "ephemeral1"
    }
```

```
]
```

run-instances 출력 예제는 예제 1을 참조하세요.

블록 디바이스 매핑에 대한 자세한 내용은 Amazon EC2 사용 설명서의 디바이스 매핑 차단을 참조하세요.

예제 4: 인스턴스를 시작하고 생성 시 태그를 추가하는 방법

다음 run-instances 예제에서는 키가 production이고 값이 webserver인 태그를 인스턴스에 추가합니다. 또한 이 명령은 키가 cost-center 이고 값이 인 태그를 생성된 cc123 모든 EBS 볼륨(이 경우 루트 볼륨)에 적용합니다.

```
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}]'
```

run-instances 출력 예제는 예제 1을 참조하세요.

예제 5: 사용자 데이터를 포함하는 인스턴스를 시작하는 방법

다음 run-instances 예제에서는 인스턴스의 구성 스크립트가 포함된 my_script.txt 파일에 사용자 데이터를 전달합니다. 스크립트는 시작할 때 실행됩니다.

```
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
```

run-instances 출력 예제는 예제 1을 참조하세요.

인스턴스 사용자 데이터에 대한 자세한 내용은 Amazon EC2 사용 설명서의 인스턴스 사용자 데이터 작업을 참조하세요.

예제 6: 성능 버스트 기능이 있는 인스턴스를 시작하는 방법

다음 `run-instances` 예제에서는 `unlimited` 크레딧 옵션을 사용하여 t2.micro 인스턴스를 시작합니다. T2 인스턴스를 시작할 때 `--credit-specification`을 지정하지 않으면 기본값은 `standard` 크레딧 옵션입니다. T3 인스턴스를 시작할 때 기본값은 `unlimited` 크레딧 옵션입니다.

```
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
```

`run-instances` 출력 예제는 예제 1을 참조하세요.

버스트 가능한 성능 인스턴스에 대한 자세한 내용은 Amazon EC2 사용 설명서의 버스트 가능한 성능 인스턴스를 참조하세요.

- 자세한 API 내용은 명령 참조RunInstances의 섹션을 참조하세요. AWS CLI

### `run-scheduled-instances`

다음 코드 예시에서는 `run-scheduled-instances`을 사용하는 방법을 보여 줍니다.

AWS CLI

예약된 인스턴스를 시작하려면

이 예제에서는 에서 지정된 예약 인스턴스를 시작합니다VPC.

명령:

```
aws ec2 run-scheduled-instances --scheduled-instance-
id sci-1234-1234-1234-1234-123456789012 --instance-count 1 --launch-
specification file://launch-specification.json
```

Launch-specification.json:

```
{
  "ImageId": "ami-12345678",
  "KeyName": "my-key-pair",
  "InstanceType": "c4.large",
  "NetworkInterfaces": [
    {
        "DeviceIndex": 0,
        "SubnetId": "subnet-12345678",
        "AssociatePublicIpAddress": true,
        "Groups": ["sg-12345678"]
    }
  ],
  "IamInstanceProfile": {
      "Name": "my-iam-role"
  }
}
```

출력:

```
{
  "InstanceIdSet": [
      "i-1234567890abcdef0"
  ]
}
```

이 예제에서는 EC2-Classic에서 지정된 예약 인스턴스를 시작합니다.

명령:

```
aws ec2 run-scheduled-instances --scheduled-instance-
id sci-1234-1234-1234-1234-123456789012 --instance-count 1 --launch-
specification file://launch-specification.json
```

Launch-specification.json:

```
{
  "ImageId": "ami-12345678",
  "KeyName": "my-key-pair",
  "SecurityGroupIds": ["sg-12345678"],
```

```
  "InstanceType": "c4.large",
  "Placement": {
    "AvailabilityZone": "us-west-2b"
  }
  "IamInstanceProfile": {
      "Name": "my-iam-role"
  }
}
```

출력:

```
{
  "InstanceIdSet": [
      "i-1234567890abcdef0"
  ]
}
```

- 자세한 API 내용은 명령 참조RunScheduledInstances의 섹션을 참조하세요. AWS CLI

### `search-local-gateway-routes`

다음 코드 예시에서는 `search-local-gateway-routes`을 사용하는 방법을 보여 줍니다.

AWS CLI

로컬 게이트웨이 라우팅 테이블에서 경로를 검색하려면

다음 `search-local-gateway-routes` 예제에서는 지정된 로컬 게이트웨이 라우팅 테이블에서 정적 경로를 검색합니다.

```
aws ec2 search-local-gateway-routes \
    --local-gateway-route-table-id lgw-rtb-059615ef7dEXAMPLE \
    --filters "Name=type,Values=static"
```

출력:

```
{
    "Route": {
        "DestinationCidrBlock": "0.0.0.0/0",
        "LocalGatewayVirtualInterfaceGroupId": "lgw-vif-grp-07145b276bEXAMPLE",
        "Type": "static",
```

```
        "State": "deleted",
        "LocalGatewayRouteTableId": "lgw-rtb-059615ef7EXAMPLE"
    }
}
```

- 자세한 API 내용은 명령 참조SearchLocalGatewayRoutes의 섹션을 참조하세요. AWS CLI

### `search-transit-gateway-multicast-groups`

다음 코드 예시에서는 `search-transit-gateway-multicast-groups`을 사용하는 방법을 보여 줍니다.

AWS CLI

하나 이상의 전송 게이트웨이 멀티캐스트 그룹을 검색하고 그룹 멤버십 정보를 반환하려면

다음 `search-transit-gateway-multicast-groups` 예제에서는 지정된 멀티캐스트 그룹의 그룹 멤버십을 반환합니다.

```
aws ec2 search-transit-gateway-multicast-groups \
    --transit-gateway-multicast-domain-id tgw-mcast-domain-000fb24d04EXAMPLE
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Transit Gateways 가이드의 멀티캐스트 그룹 관리를 참조하세요.

- 자세한 API 내용은 명령 참조SearchTransitGatewayMulticastGroups의 섹션을 참조하세요. AWS CLI

### `search-transit-gateway-routes`

다음 코드 예시에서는 `search-transit-gateway-routes`을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 전송 게이트웨이 라우팅 테이블에서 경로를 검색하려면

다음 `search-transit-gateway-routes` 예제에서는 지정된 라우팅 테이블에 유형인 모든 라우팅`static`을 반환합니다.

```
aws ec2 search-transit-gateway-routes \
    --transit-gateway-route-table-id tgw-rtb-0a823edbdeEXAMPLE \
    --filters "Name=type,Values=static"
```

출력:

```
{
    "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
}
```

자세한 내용은 Transit Gateways 가이드의 Transit Gateway route table을 참조하세요.

- 자세한 API 내용은 명령 참조SearchTransitGatewayRoutes의 섹션을 참조하세요. AWS CLI

### `send-diagnostic-interrupt`

다음 코드 예시에서는 `send-diagnostic-interrupt`을 사용하는 방법을 보여 줍니다.

AWS CLI

진단 인터럽트를 보내려면

다음 `send-diagnostic-interrupt` 예제에서는 지정된 인스턴스에 진단 인터럽트를 보냅니다.

```
aws ec2 send-diagnostic-interrupt \
    --instance-id i-1234567890abcdef0
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조SendDiagnosticInterrupt의 섹션을 참조하세요. AWS CLI

### `start-instances`

다음 코드 예시에서는 `start-instances`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon EC2 인스턴스를 시작하려면

이 예제에서는 지정된 Amazon EBS지원 인스턴스를 시작합니다.

명령:

```
aws ec2 start-instances --instance-ids i-1234567890abcdef0
```

출력:

```
{
    "StartingInstances": [
        {
            "InstanceId": "i-1234567890abcdef0",
            "CurrentState": {
                "Code": 0,
                "Name": "pending"
            },
            "PreviousState": {
                "Code": 80,
                "Name": "stopped"
            }
        }
    ]
}
```

자세한 내용은 Amazon Elastic Compute Cloud 사용 설명서에서 인스턴스 중지 및 시작을 참조하세요.

- 자세한 API 내용은 명령 참조StartInstances의 섹션을 참조하세요. AWS CLI

### `start-network-insights-access-scope-analysis`

다음 코드 예시에서는 `start-network-insights-access-scope-analysis`을 사용하는 방법을 보여 줍니다.

AWS CLI

Network Insights 액세스 범위 분석을 시작하려면

다음 `start-network-insights-access-scope-analysis` 예제에서는 AWS 계정에서 범위 분석을 시작합니다.

```
aws ec2 start-network-insights-access-scope-analysis \
    --region us-east-1 \
    --network-insights-access-scope-id nis-123456789111
```

출력:

```
{
```

```
    "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"
    }
}
```

자세한 내용은 Network Access Analyzer 안내서의 를 사용하여 AWS CLI Network Access Analyzer 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조StartNetworkInsightsAccessScopeAnalysis의 섹션을 참조하세요. AWS CLI

### `start-network-insights-analysis`

다음 코드 예시에서는 `start-network-insights-analysis`을 사용하는 방법을 보여 줍니다.

AWS CLI

경로를 분석하려면

다음 `start-network-insights-analysis` 예제에서는 소스와 대상 간의 경로를 분석합니다. 경로 분석 결과를 보려면 `describe-network-insights-analyses` 명령을 사용합니다.

```
aws ec2 start-network-insights-analysis \
    --network-insights-path-id nip-0b26f224f1d131fa8
```

출력:

```
{
    "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"
    }
```

```
}
```

자세한 내용은 Reachability Analyzer 가이드의 시작하기 AWS CLI를 참조하세요.

- 자세한 API 내용은 명령 참조StartNetworkInsightsAnalysis의 섹션을 참조하세요. AWS CLI

### `start-vpc-endpoint-service-private-dns-verification`

다음 코드 예시에서는 `start-vpc-endpoint-service-private-dns-verification`을 사용하는 방법을 보여 줍니다.

AWS CLI

DNS 확인 프로세스를 시작하려면

다음 `start-vpc-endpoint-service-private-dns-verification` 예제에서는 지정된 엔드포인트 서비스에 대한 DNS 확인 프로세스를 시작합니다.

```
aws ec2 start-vpc-endpoint-service-private-dns-verification \
    --service-id vpce-svc-071afff70666e61e0
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS PrivateLink 사용 설명서의 DNS 이름 관리를 참조하세요.

- 자세한 API 내용은 명령 참조StartVpcEndpointServicePrivateDnsVerification의 섹션을 참조하세요. AWS CLI

### `stop-instances`

다음 코드 예시에서는 `stop-instances`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: Amazon EC2 인스턴스를 중지하려면

다음 `stop-instances` 예제에서는 지정된 Amazon EBS지원 인스턴스를 중지합니다.

```
aws ec2 stop-instances \
    --instance-ids i-1234567890abcdef0
```

출력:

```
{
    "StoppingInstances": [
        {
            "InstanceId": "i-1234567890abcdef0",
            "CurrentState": {
                "Code": 64,
                "Name": "stopping"
            },
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
```

자세한 내용은 Amazon Elastic Compute Cloud 사용 설명서에서 인스턴스 중지 및 시작을 참조하세요.

예제 2: Amazon EC2 인스턴스 최대 절전 모드 해제

다음 `stop-instances` 예제는 인스턴스가 최대 절전 모드에 대해 활성화되어 있고 최대 절전 모드 사전 조건을 충족하는 경우 Amazon EBS지원 인스턴스를 최대 절전 모드로 전환합니다. 인스턴스가 최대 절전 모드로 전환된 후에 인스턴스가 중지됩니다.

```
aws ec2 stop-instances \
    --instance-ids i-1234567890abcdef0 \
    --hibernate
```

출력:

```
{
    "StoppingInstances": [
        {
            "CurrentState": {
                "Code": 64,
                "Name": "stopping"
            },
            "InstanceId": "i-1234567890abcdef0",
```

```
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
```

자세한 내용은 Amazon Elastic Compute Cloud 사용 설명서에서 온디맨드 Linux 인스턴스를 최대 절전 모드로 전환을 참조하세요.

- 자세한 API 내용은 명령 참조StopInstances의 섹션을 참조하세요. AWS CLI

### `terminate-client-vpn-connections`

다음 코드 예시에서는 `terminate-client-vpn-connections`을 사용하는 방법을 보여 줍니다.

AWS CLI

클라이언트 VPN 엔드포인트에 대한 연결을 종료하려면

다음 `terminate-client-vpn-connections` 예제에서는 클라이언트 VPN 엔드포인트에 대한 지정된 연결을 종료합니다.

```
aws ec2 terminate-client-vpn-connections \
    --client-vpn-endpoint-id vpn-endpoint-123456789123abcde \
    --connection-id cvpn-connection-04edd76f5201e0cb8
```

출력:

```
{
    "ClientVpnEndpointId": "vpn-endpoint-123456789123abcde",
    "ConnectionStatuses": [
        {
            "ConnectionId": "cvpn-connection-04edd76f5201e0cb8",
            "PreviousStatus": {
                "Code": "active"
            },
            "CurrentStatus": {
                "Code": "terminating"
            }
        }
```

```
    ]
}
```

자세한 내용은 클라이언트 관리자 안내서의 클라이언트 연결을 참조하세요. AWS VPN

- 자세한 API 내용은 명령 참조TerminateClientVpnConnections의 섹션을 참조하세요. AWS CLI

### `terminate-instances`

다음 코드 예시에서는 `terminate-instances`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon EC2 인스턴스를 종료하려면

이 예제에서는 지정된 인스턴스를 종료합니다.

명령:

```
aws ec2 terminate-instances --instance-ids i-1234567890abcdef0
```

출력:

```
{
    "TerminatingInstances": [
        {
            "InstanceId": "i-1234567890abcdef0",
            "CurrentState": {
                "Code": 32,
                "Name": "shutting-down"
            },
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
```

자세한 내용은 AWS 명령줄 인터페이스 사용 설명서의 Amazon EC2 인스턴스 사용을 참조하세요.

- 자세한 API 내용은 명령 참조TerminateInstances의 섹션을 참조하세요. AWS CLI

### unassign-ipv6-addresses

다음 코드 예시에서는 unassign-ipv6-addresses을 사용하는 방법을 보여 줍니다.

AWS CLI

네트워크 인터페이스에서 IPv6 주소 할당을 취소하려면

이 예제에서는 지정된 네트워크 인터페이스에서 지정된 IPv6 주소를 할당 취소합니다.

명령:

```
aws ec2 unassign-ipv6-addresses --ipv6-
addresses 2001:db8:1234:1a00:3304:8879:34cf:4071 --network-interface-id eni-23c49b68
```

출력:

```
{
  "NetworkInterfaceId": "eni-23c49b68",
  "UnassignedIpv6Addresses": [
      "2001:db8:1234:1a00:3304:8879:34cf:4071"
  ]
}
```

- API 자세한 내용은 AWS CLI 명령 참조의 UnassignIpv6Addresses를 참조하세요.

### unassign-private-ip-addresses

다음 코드 예시에서는 unassign-private-ip-addresses을 사용하는 방법을 보여 줍니다.

AWS CLI

네트워크 인터페이스에서 보조 프라이빗 IP 주소 할당을 취소하려면

이 예제에서는 지정된 네트워크 인터페이스에서 지정된 프라이빗 IP 주소를 할당 취소합니다. 이 명령이 성공하면 출력이 반환되지 않습니다.

명령:

```
aws ec2 unassign-private-ip-addresses --network-interface-id eni-e5aa89a3 --private-
ip-addresses 10.0.0.82
```

- 자세한 API 내용은 명령 참조UnassignPrivateIpAddresses의 섹션을 참조하세요. AWS CLI

### unassign-private-nat-gateway-address

다음 코드 예시에서는 unassign-private-nat-gateway-address을 사용하는 방법을 보여 줍니다.

AWS CLI

프라이빗 NAT 게이트웨이에서 프라이빗 IP 주소 할당을 취소하려면

다음 unassign-private-nat-gateway-address 예제에서는 지정된 프라이빗 NAT 게이트웨이에서 지정된 IP 주소를 할당 취소합니다.

```
aws ec2 unassign-private-nat-gateway-address \
    --nat-gateway-id nat-1234567890abcdef0 \
    --private-ip-addresses 10.0.20.197
```

출력:

```
{
    "NatGatewayId": "nat-0ee3edd182361f662",
    "NatGatewayAddresses": [
        {
            "NetworkInterfaceId": "eni-0065a61b324d1897a",
            "PrivateIp": "10.0.20.197",
            "IsPrimary": false,
            "Status": "unassigning"
        }
    ]
}
```

자세한 내용은 Amazon VPC 사용 설명서의 NAT 게이트웨이를 참조하세요.

- 자세한 API 내용은 명령 참조UnassignPrivateNatGatewayAddress의 섹션을 참조하세요. AWS CLI

### unlock-snapshot

다음 코드 예시에서는 unlock-snapshot을 사용하는 방법을 보여 줍니다.

AWS CLI

스냅샷을 잠금 해제하려면

다음 unlock-snapshot 예제에서는 지정된 스냅샷을 잠금 해제합니다.

```
aws ec2 unlock-snapshot \
    --snapshot-id snap-0b5e733b4a8df6e0d
```

출력:

```
{
    "SnapshotId": "snap-0b5e733b4a8df6e0d"
}
```

자세한 내용은 Amazon EBS 사용 설명서의 스냅샷 잠금을 참조하세요.

- 자세한 API 내용은 명령 참조UnlockSnapshot의 섹션을 참조하세요. AWS CLI

### unmonitor-instances

다음 코드 예시에서는 unmonitor-instances을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스에 대한 세부 모니터링을 비활성화하는 방법

이 예제 명령은 지정된 인스턴스에 대한 세부 모니터링을 비활성화합니다.

명령:

```
aws ec2 unmonitor-instances --instance-ids i-1234567890abcdef0
```

출력:

```
{
  "InstanceMonitorings": [
      {
          "InstanceId": "i-1234567890abcdef0",
          "Monitoring": {
              "State": "disabling"
          }
```

```
        }
    ]
}
```

- 자세한 API 내용은 명령 참조UnmonitorInstances의 섹션을 참조하세요. AWS CLI

### update-security-group-rule-descriptions-egress

다음 코드 예시에서는 update-security-group-rule-descriptions-egress을 사용하는 방법을 보여 줍니다.

AWS CLI

아웃바운드 보안 그룹 규칙에 대한 설명을 업데이트하려면

다음 update-security-group-rule-descriptions-egress 예제에서는 지정된 포트 및 IPv4 주소 범위에 대한 보안 그룹 규칙에 대한 설명을 업데이트합니다. 'Outbound HTTP access to server 2' 설명은 규칙에 대한 기존 설명을 대체합니다.

```
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"}]
```

출력:

```
{
    "Return": true
}
```

자세한 내용은 Amazon EC2 사용 설명서의 보안 그룹 규칙을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateSecurityGroupRuleDescriptionsEgress의 섹션을 참조하세요. AWS CLI

### update-security-group-rule-descriptions-ingress

다음 코드 예시에서는 update-security-group-rule-descriptions-ingress을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 인바운드 보안 그룹 규칙에 대한 설명을 CIDR 소스로 업데이트하려면

다음 update-security-group-rule-descriptions-ingress 예제에서는 지정된 포트 및 IPv4 주소 범위에 대한 보안 그룹 규칙에 대한 설명을 업데이트합니다. 'SSH access from ABC office' 설명은 규칙에 대한 기존 설명을 대체합니다.

```
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"}]'
```

출력:

```
{
    "Return": true
}
```

자세한 내용은 Amazon EC2 사용 설명서의 보안 그룹 규칙을 참조하세요.

예제 2: 인바운드 보안 그룹 규칙의 설명을 접두사 목록 소스로 업데이트하는 방법

다음 update-security-group-rule-descriptions-ingress 예제에서는 지정된 포트 및 접두사 목록에 대한 보안 그룹 규칙에 대한 설명을 업데이트합니다. 'SSH access from ABC office' 설명은 규칙에 대한 기존 설명을 대체합니다.

```
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"}]'
```

출력:

```
{
    "Return": true
}
```

자세한 내용은 Amazon EC2 사용 설명서의 보안 그룹 규칙을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateSecurityGroupRuleDescriptionsIngress의 섹션을 참조하세요. AWS CLI

### `withdraw-byoip-cidr`

다음 코드 예시에서는 `withdraw-byoip-cidr`을 사용하는 방법을 보여 줍니다.

AWS CLI

주소 범위 광고를 중지하려면

다음 `withdraw-byoip-cidr` 예제에서는 지정된 주소 범위의 광고를 중지합니다.

```
aws ec2 withdraw-byoip-cidr
    --cidr 203.0.113.25/24
```

출력:

```
{
    "ByoipCidr": {
        "Cidr": "203.0.113.25/24",
        "StatusMessage": "ipv4pool-ec2-1234567890abcdef0",
        "State": "advertised"
    }
}
```

- 자세한 API 내용은 명령 참조WithdrawByoipCidr의 섹션을 참조하세요. AWS CLI

## 를 사용한 Amazon EC2 Instance Connect 예제 AWS CLI

다음 코드 예제에서는 Amazon EC2 Instance Connect와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### send-ssh-public-key

다음 코드 예시에서는 send-ssh-public-key을 사용하는 방법을 보여 줍니다.

AWS CLI

SSH 퍼블릭 키를 인스턴스로 보내려면

다음 send-ssh-public-key 예제에서는 지정된 SSH 퍼블릭 키를 지정된 인스턴스로 보냅니다. 키는 지정된 사용자를 인증하는 데 사용됩니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조SendSshPublicKey의 섹션을 참조하세요. AWS CLI

## 를 사용한 Amazon ECR 예제 AWS CLI

다음 코드 예제에서는 Amazon 와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다ECR.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

작업

### batch-check-layer-availability

다음 코드 예시에서는 batch-check-layer-availability을 사용하는 방법을 보여 줍니다.

AWS CLI

계층의 가용성을 확인하려면

다음 batch-check-layer-availability 예제에서는 cluster-autoscaler리포지토리에 다이제스트가 있는 계층의 가용성sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed을 확인합니다.

```
aws ecr batch-check-layer-availability \
    --repository-name cluster-autoscaler \
    --layer-
digests sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed
```

출력:

```
{
    "layers": [
        {
            "layerDigest":
 "sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed",
            "layerAvailability": "AVAILABLE",
            "layerSize": 2777,
            "mediaType": "application/vnd.docker.container.image.v1+json"
        }
    ],
    "failures": []
}
```

- 자세한 API 내용은 명령 참조BatchCheckLayerAvailability의 섹션을 참조하세요. AWS CLI

### batch-delete-image

다음 코드 예시에서는 batch-delete-image을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 이미지를 삭제하려면

다음 batch-delete-image 예제에서는 계정의 기본 레지스트리precise에서 지정된 리포지토리에 태그가 있는 이미지를 삭제합니다.

```
aws ecr batch-delete-image \
    --repository-name ubuntu \
    --image-ids imageTag=precise
```

출력:

```
{
    "failures": [],
    "imageIds": [
        {
            "imageTag": "precise",
            "imageDigest":
 "sha256:19665f1e6d1e504117a1743c0a3d3753086354a38375961f2e665416ef4b1b2f"
        }
    ]
}
```

예제 2: 여러 이미지를 삭제하려면

다음 batch-delete-image 예제에서는 지정된 리포지토리team1에서 prod 및 로 태그가 지정된 모든 이미지를 삭제합니다.

```
aws ecr batch-delete-image \
    --repository-name MyRepository \
    --image-ids imageTag=prod imageTag=team1
```

출력:

```
{
    "imageIds": [
        {
            "imageDigest": "sha256:123456789012",
            "imageTag": "prod"
        },
        {
```

```
            "imageDigest": "sha256:567890121234",
            "imageTag": "team1"
        }
    ],
    "failures": []
}
```

자세한 내용은 Amazon ECR 사용 설명서의 이미지 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조BatchDeleteImage의 섹션을 참조하세요. AWS CLI

### batch-get-image

다음 코드 예시에서는 batch-get-image을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 이미지 가져오기

다음 batch-get-image 예제에서는 계정의 기본 레지스트리cluster-autoscaler에서 라는 리포지토리v1.13.6에 태그가 있는 이미지를 가져옵니다.

```
aws ecr batch-get-image \
    --repository-name cluster-autoscaler \
    --image-ids imageTag=v1.13.6
```

출력:

```
{
    "images": [
        {
            "registryId": "012345678910",
            "repositoryName": "cluster-autoscaler",
            "imageId": {
                "imageDigest":
 "sha256:4a1c6567c38904384ebc64e35b7eeddd8451110c299e3368d2210066487d97e5",
                "imageTag": "v1.13.6"
            },
            "imageManifest": "{\n   \"schemaVersion\": 2,\n
 \"mediaType\": \"application/vnd.docker.distribution.manifest.v2+json
\",\n   \"config\": {\n      \"mediaType\": \"application/
vnd.docker.container.image.v1+json\",\n      \"size\": 2777,\n      \"digest
```

```
\": \"sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed
\"\n   },\n   \"layers\": [\n      {\n         \"mediaType
\": \"application/vnd.docker.image.rootfs.diff.tar.gzip
\",\n         \"size\": 17743696,\n         \"digest\":
 \"sha256:39fafc05754f195f134ca11ecdb1c9a691ab0848c697fffeb5a85f900caaf6e1\"\n
      },\n      {\n         \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\",\n         \"size\": 2565026,\n
 \"digest\":
 \"sha256:8c8a779d3a537b767ae1091fe6e00c2590afd16767aa6096d1b318d75494819f
\"\n      },\n      {\n         \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\",\n         \"size\": 28005981,\n
 \"digest\":
 \"sha256:c44ba47496991c9982ee493b47fd25c252caabf2b4ae7dd679c9a27b6a3c8fb7\"\n
      },\n      {\n         \"mediaType\": \"application/
vnd.docker.image.rootfs.diff.tar.gzip\",\n         \"size\": 775,\n         \"digest
\": \"sha256:e2c388b44226544363ca007be7b896bcce1baebea04da23cbd165eac30be650f\"\n
   }\n   ]\n}"
        }
    ],
    "failures": []
}
```

예제 2: 여러 이미지를 가져오려면

다음 batch-get-image 예제에서는 지정된 리포지토리team1에 prod 및 로 태그가 지정된 모든 이미지의 세부 정보를 표시합니다.

```
aws ecr batch-get-image \
    --repository-name MyRepository \
    --image-ids imageTag=prod imageTag=team1
```

출력:

```
{
    "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": []
}
```

자세한 내용은 Amazon ECR 사용 설명서의 이미지를 참조하세요.

- 자세한 API 내용은 명령 참조BatchGetImage의 섹션을 참조하세요. AWS CLI

### complete-layer-upload

다음 코드 예시에서는 complete-layer-upload을 사용하는 방법을 보여 줍니다.

AWS CLI

이미지 계층 업로드를 완료하려면

다음 complete-layer-upload 예제에서는 layer-test리포지토리에 이미지 계층 업로드를 완료합니다.

```
aws ecr complete-layer-upload \
    --repository-name layer-test \
    --upload-id 6cb64b8a-9378-0e33-2ab1-b780fab8a9e9 \
    --layer-digests 6cb64b8a-9378-0e33-2ab1-
b780fab8a9e9:48074e6d3a68b39aad8ccc002cdad912d4148c0f92b3729323e
```

출력:

```
{
    "uploadId": "6cb64b8a-9378-0e33-2ab1-b780fab8a9e9",
    "layerDigest":
 "sha256:9a77f85878aa1906f2020a0ecdf7a7e962d57e882250acd773383224b3fe9a02",
    "repositoryName": "layer-test",
    "registryId": "130757420319"
```

```
}
```

- 자세한 API 내용은 명령 참조CompleteLayerUpload의 섹션을 참조하세요. AWS CLI

### `create-repository`

다음 코드 예시에서는 `create-repository`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 리포지토리 생성

다음 `create-repository` 예제에서는 계정의 기본 레지스트리에서 지정된 네임스페이스 내에 리포지토리를 생성합니다.

```
aws ecr create-repository \
    --repository-name project-a/sample-repo
```

출력:

```
{
    "repository": {
        "registryId": "123456789012",
        "repositoryName": "project-a/sample-repo",
        "repositoryArn": "arn:aws:ecr:us-west-2:123456789012:repository/project-a/
sample-repo"
    }
}
```

자세한 내용은 Amazon ECR 사용 설명서의 리포지토리 생성을 참조하세요.

예제 2: 이미지 태그 변경 불가능으로 구성된 리포지토리 생성

다음 `create-repository` 예제에서는 계정의 기본 레지스트리에서 태그 불변성을 위해 구성된 리포지토리를 생성합니다.

```
aws ecr create-repository \
    --repository-name project-a/sample-repo \
    --image-tag-mutability IMMUTABLE
```

출력:

```
{
    "repository": {
        "registryId": "123456789012",
        "repositoryName": "project-a/sample-repo",
        "repositoryArn": "arn:aws:ecr:us-west-2:123456789012:repository/project-a/
sample-repo",
        "imageTagMutability": "IMMUTABLE"
    }
}
```

자세한 내용은 Amazon ECR 사용 설명서의 이미지 태그 이동성을 참조하세요.

예제 3: 스캔 구성으로 구성된 리포지토리 생성

다음 `create-repository` 예제에서는 계정의 기본 레지스트리에서 이미지 푸시에 대한 취약성 스캔을 수행하도록 구성된 리포지토리를 생성합니다.

```
aws ecr create-repository \
    --repository-name project-a/sample-repo \
    --image-scanning-configuration scanOnPush=true
```

출력:

```
{
    "repository": {
        "registryId": "123456789012",
        "repositoryName": "project-a/sample-repo",
        "repositoryArn": "arn:aws:ecr:us-west-2:123456789012:repository/project-a/
sample-repo",
        "imageScanningConfiguration": {
            "scanOnPush": true
        }
    }
}
```

자세한 내용은 Amazon ECR 사용 설명서의 이미지 스캔을 참조하세요.

- 자세한 API 내용은 명령 참조CreateRepository의 섹션을 참조하세요. AWS CLI

### `delete-lifecycle-policy`

다음 코드 예시에서는 `delete-lifecycle-policy`을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리의 수명 주기 정책을 삭제하려면

다음 delete-lifecycle-policy 예제에서는 hello-world리포지토리의 수명 주기 정책을 삭제합니다.

```
aws ecr delete-lifecycle-policy \
    --repository-name hello-world
```

출력:

```
{
    "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
}
```

- 자세한 API 내용은 명령 참조DeleteLifecyclePolicy의 섹션을 참조하세요. AWS CLI

### delete-repository-policy

다음 코드 예시에서는 delete-repository-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리에 대한 리포지토리 정책을 삭제하려면

다음 delete-repository-policy 예제에서는 리포지토리에 대한 리포지cluster-autoscaler토리 정책을 삭제합니다.

```
aws ecr delete-repository-policy \
    --repository-name cluster-autoscaler
```

출력:

```
{
```

```
    "registryId": "012345678910",
    "repositoryName": "cluster-autoscaler",
    "policyText": "{\n  \"Version\" : \"2008-10-17\",\n  \"Statement\" : [ {\n
 \"Sid\" : \"allow public pull\",\n    \"Effect\" : \"Allow\",\n    \"Principal\" :
 \"*\",\n    \"Action\" : [ \"ecr:BatchCheckLayerAvailability\", \"ecr:BatchGetImage
\", \"ecr:GetDownloadUrlForLayer\" ]\n  } ]\n}"
}
```

- 자세한 API 내용은 명령 참조DeleteRepositoryPolicy의 섹션을 참조하세요. AWS CLI

### delete-repository

다음 코드 예시에서는 delete-repository을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리를 삭제하려면

다음 delete-repository 예제 명령 힘은 계정의 기본 레지스트리에서 지정된 리포지토리를 삭제합니다. 리포지토리에 이미지가 포함된 경우 --force 플래그가 필요합니다.

```
aws ecr delete-repository \
    --repository-name ubuntu \
    --force
```

출력:

```
{
    "repository": {
        "registryId": "123456789012",
        "repositoryName": "ubuntu",
        "repositoryArn": "arn:aws:ecr:us-west-2:123456789012:repository/ubuntu"
    }
}
```

자세한 내용은 Amazon ECR 사용 설명서의 리포지토리 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteRepository의 섹션을 참조하세요. AWS CLI

### describe-image-scan-findings

다음 코드 예시에서는 describe-image-scan-findings을 사용하는 방법을 보여 줍니다.

AWS CLI

이미지의 스캔 결과를 설명하려면

다음 `describe-image-scan-findings` 예제에서는 계정의 기본 레지스트리에 지정된 리포지토리에서 이미지 다이제스트를 사용하여 이미지에 대한 이미지 스캔 결과를 반환합니다.

```
aws ecr describe-image-scan-findings \
    --repository-name sample-repo \
    --image-
id imageDigest=sha256:74b2c688c700ec95a93e478cdb959737c148df3fbf5ea706abe0318726e885e6
```

출력:

```
{
    "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."
    }
}
```

자세한 내용은 Amazon ECR 사용 설명서의 이미지 스캔을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeImageScanFindings의 섹션을 참조하세요. AWS CLI

### describe-images

다음 코드 예시에서는 describe-images을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리의 이미지를 설명하려면

아래 describe-images 예제에서는 태그가 인 cluster-autoscaler리포지토리의 이미지에 대한 세부 정보를 표시합니다v1.13.6.

```
aws ecr describe-images \
    --repository-name cluster-autoscaler \
    --image-ids imageTag=v1.13.6
```

출력:

```
{
    "imageDetails": [
```

```
        {
            "registryId": "012345678910",
            "repositoryName": "cluster-autoscaler",
            "imageDigest":
 "sha256:4a1c6567c38904384ebc64e35b7eeddd8451110c299e3368d2210066487d97e5",
            "imageTags": [
                "v1.13.6"
            ],
            "imageSizeInBytes": 48318255,
            "imagePushedAt": 1565128275.0
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeImages의 섹션을 참조하세요. AWS CLI

### `describe-repositories`

다음 코드 예시에서는 `describe-repositories`을 사용하는 방법을 보여 줍니다.

AWS CLI

레지스트리의 리포지토리를 설명하는 방법

이 예시에서는 계정의 기본 레지스트리에 있는 리포지토리를 설명합니다.

명령:

```
aws ecr describe-repositories
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeRepositories의 섹션을 참조하세요. AWS CLI

### get-authorization-token

다음 코드 예시에서는 get-authorization-token을 사용하는 방법을 보여 줍니다.

AWS CLI

기본 레지스트리에 대한 권한 부여 토큰을 가져오려면

다음 get-authorization-token 예제 명령은 기본 레지스트리에 대한 권한 부여 토큰을 가져옵니다.

```
aws ecr get-authorization-token
```

출력:

```
{
    "authorizationData": [
        {
            "authorizationToken": "QVdTOkN...",
            "expiresAt": 1448875853.241,
            "proxyEndpoint": "https://123456789012.dkr.ecr.us-west-2.amazonaws.com"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetAuthorizationToken의 섹션을 참조하세요. AWS CLI

### get-download-url-for-layer

다음 코드 예시에서는 get-download-url-for-layer을 사용하는 방법을 보여 줍니다.

AWS CLI

URL 계층 다운로드

다음 get-download-url-for-layer 예제에서는 cluster-autoscaler리포지토리에 다이제스트가 있는 계층URL의 다운로드sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed를 보여줍니다.

```
aws ecr get-download-url-for-layer \
    --repository-name cluster-autoscaler  \
    --layer-
digest sha256:6171c7451a50945f8ddd72f7732cc04d7a0d1f48138a426b2e64387fdeb834ed
```

출력:

```
{
    "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"
}
```

- 자세한 API 내용은 명령 참조GetDownloadUrlForLayer의 섹션을 참조하세요. AWS CLI

### get-lifecycle-policy-preview

다음 코드 예시에서는 get-lifecycle-policy-preview을 사용하는 방법을 보여 줍니다.

AWS CLI

수명 주기 정책 미리 보기에 대한 세부 정보를 검색하려면

다음 get-lifecycle-policy-preview 예제에서는 계정의 기본 레지스트리에 지정된 리포지토리에 대한 수명 주기 정책 미리 보기 결과를 검색합니다.

명령:

```
aws ecr get-lifecycle-policy-preview \
    --repository-name "project-a/amazon-ecs-sample"
```

출력:

```
{
    "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}\n",
    "status": "COMPLETE",
    "previewResults": [],
    "summary": {
        "expiringImageTotalCount": 0
    }
}
```

자세한 내용은 Amazon ECR 사용 설명서의 수명 주기 정책을 참조하세요.

- 자세한 API 내용은 명령 참조GetLifecyclePolicyPreview의 섹션을 참조하세요. AWS CLI

### get-lifecycle-policy

다음 코드 예시에서는 get-lifecycle-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

수명 주기 정책을 검색하려면

다음 get-lifecycle-policy 예제에서는 계정의 기본 레지스트리에 지정된 리포지토리의 수명 주기 정책에 대한 세부 정보를 표시합니다.

```
aws ecr get-lifecycle-policy \
    --repository-name "project-a/amazon-ecs-sample"
```

출력:

```
{
     "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
}
```

자세한 내용은 Amazon ECR 사용 설명서의 수명 주기 정책을 참조하세요.

- 자세한 API 내용은 명령 참조GetLifecyclePolicy의 섹션을 참조하세요. AWS CLI

### get-login-password

다음 코드 예시에서는 get-login-password을 사용하는 방법을 보여 줍니다.

AWS CLI

레지스트리에 인증할 암호를 검색하려면

다음은 IAM 보안 주체가 액세스할 수 있는 Amazon ECR 레지스트리에 인증하는 데 선택한 컨테이너 클라이언트와 함께 사용할 수 있는 암호를 get-login-password 표시합니다.

```
aws ecr get-login-password
```

출력:

```
<password>
```

Docker 와 함께 사용하려면 get-login-password 명령의 출력을 docker login 명령에 CLI파이프합니다. 암호를 검색할 때 Amazon ECR 레지스트리가 존재하는 리전과 동일한 리전을 지정해야 합니다.

```
aws ecr get-login-password \
    --region <region>
 \
| docker login \
    --username AWS \
    --password-stdin <aws_account_id>.dkr.ecr.<region>.amazonaws.com
```

자세한 내용은 Amazon ECR 사용 설명서의 레지스트리 인증을 참조하세요.

- 자세한 API 내용은 명령 참조GetLoginPassword의 섹션을 참조하세요. AWS CLI

### get-login

다음 코드 예시에서는 get-login을 사용하는 방법을 보여 줍니다.

AWS CLI

기본 레지스트리에 대한 Docker 로그인 명령을 검색하려면

이 예제에서는 기본 Amazon ECR 레지스트리에 로그인하는 데 사용할 수 있는 명령을 인쇄합니다.

명령:

```
aws ecr get-login
```

출력:

```
docker login -u AWS -p <password> -e none https://
<aws_account_id>.dkr.ecr.<region>.amazonaws.com
```

다른 계정의 레지스트리에 로그인하려면

이 예제에서는 다른 계정과 연결된 Amazon ECR 레지스트리에 로그인하는 데 사용할 수 있는 명령을 하나 이상 인쇄합니다.

명령:

```
aws ecr get-login --registry-ids 012345678910 023456789012
```

출력:

```
docker login -u <username> -p <token-1> -e none <endpoint-1>
docker login -u <username> -p <token-2> -e none <endpoint-2>
```

- 자세한 API 내용은 명령 참조GetLogin의 섹션을 참조하세요. AWS CLI

### get-repository-policy

다음 코드 예시에서는 get-repository-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리에 대한 리포지토리 정책을 검색하려면

다음 get-repository-policy 예제에서는 리포지토리의 리cluster-autoscaler포지토리 정책에 대한 세부 정보를 보여줍니다.

```
aws ecr get-repository-policy \
    --repository-name cluster-autoscaler
```

출력:

```
{
    "registryId": "012345678910",
    "repositoryName": "cluster-autoscaler",
    "policyText": "{\n  \"Version\" : \"2008-10-17\",\n  \"Statement\" : [ {\n
 \"Sid\" : \"allow public pull\",\n    \"Effect\" : \"Allow\",\n    \"Principal\" :
 \"*\",\n    \"Action\" : [ \"ecr:BatchCheckLayerAvailability\", \"ecr:BatchGetImage
\", \"ecr:GetDownloadUrlForLayer\" ]\n  } ]\n}"
}
```

- 자세한 API 내용은 명령 참조GetRepositoryPolicy의 섹션을 참조하세요. AWS CLI

### initiate-layer-upload

다음 코드 예시에서는 initiate-layer-upload을 사용하는 방법을 보여 줍니다.

AWS CLI

이미지 계층 업로드를 시작하려면

다음 initiate-layer-upload 예제에서는 layer-test리포지토리에 이미지 계층 업로드를 시작합니다.

```
aws ecr initiate-layer-upload \
    --repository-name layer-test
```

출력:

```
{
    "partSize": 10485760,
    "uploadId": "6cb64b8a-9378-0e33-2ab1-b780fab8a9e9"
}
```

- 자세한 API 내용은 명령 참조InitiateLayerUpload의 섹션을 참조하세요. AWS CLI

### list-images

다음 코드 예시에서는 list-images을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리의 이미지를 나열하는 방법

다음 list-images 예시에서는 cluster-autoscaler 리포지토리의 이미지 목록을 표시합니다.

```
aws ecr list-images \
    --repository-name cluster-autoscaler
```

출력:

```
{
    "imageIds": [
        {
            "imageDigest":
 "sha256:99c6fb4377e9a420a1eb3b410a951c9f464eff3b7dbc76c65e434e39b94b6570",
            "imageTag": "v1.13.8"
        },
        {
            "imageDigest":
 "sha256:99c6fb4377e9a420a1eb3b410a951c9f464eff3b7dbc76c65e434e39b94b6570",
            "imageTag": "v1.13.7"
        },
        {
            "imageDigest":
 "sha256:4a1c6567c38904384ebc64e35b7eeddd8451110c299e3368d2210066487d97e5",
            "imageTag": "v1.13.6"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListImages의 섹션을 참조하세요. AWS CLI

### list-tags-for-resource

다음 코드 예시에서는 list-tags-for-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리의 태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 `hello-world`리포지토리와 연결된 태그 목록을 표시합니다.

```
aws ecr list-tags-for-resource \
    --resource-arn arn:aws:ecr:us-west-2:012345678910:repository/hello-world
```

출력:

```
{
    "tags": [
        {
            "Key": "Stage",
            "Value": "Integ"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### `put-image-scanning-configuration`

다음 코드 예시에서는 `put-image-scanning-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리의 이미지 스캔 구성을 업데이트하려면

다음 `put-image-scanning-configuration` 예제에서는 지정된 리포지토리에 대한 이미지 스캔 구성을 업데이트합니다.

```
aws ecr put-image-scanning-configuration \
    --repository-name sample-repo \
    --image-scanning-configuration scanOnPush=true
```

출력:

```
{
```

```
    "registryId": "012345678910",
    "repositoryName": "sample-repo",
    "imageScanningConfiguration": {
      "scanOnPush": true
    }
}
```

자세한 내용은 Amazon ECR 사용 설명서의 이미지 스캔을 참조하세요.

- 자세한 API 내용은 명령 참조PutImageScanningConfiguration의 섹션을 참조하세요. AWS CLI

### put-image-tag-mutability

다음 코드 예시에서는 put-image-tag-mutability을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리의 이미지 태그 변경 가능성 설정을 업데이트하려면

다음 put-image-tag-mutability 예제에서는 태그 불변성을 위해 지정된 리포지토리를 구성합니다. 이렇게 하면 리포지토리 내의 모든 이미지 태그를 덮어쓰지 못합니다.

```
aws ecr put-image-tag-mutability \
    --repository-name hello-repository \
    --image-tag-mutability IMMUTABLE
```

출력:

```
{
    "registryId": "012345678910",
    "repositoryName": "sample-repo",
    "imageTagMutability": "IMMUTABLE"
}
```

자세한 내용은 Amazon ECR 사용 설명서의 이미지 태그 이동성을 참조하세요.

- 자세한 API 내용은 명령 참조PutImageTagMutability의 섹션을 참조하세요. AWS CLI

### put-image

다음 코드 예시에서는 put-image을 사용하는 방법을 보여 줍니다.

AWS CLI

매니페스트를 사용하여 이미지에 다시 태그를 지정하려면

다음 put-image 예제에서는 기존 이미지 매니페스트를 사용하여 hello-world리포지토리에 새 태그를 생성합니다.

```
aws ecr put-image \
    --repository-name hello-world \
    --image-tag 2019.08 \
    --image-manifest file://hello-world.manifest.json
```

hello-world.manifest.json의 콘텐츠:

```
{
    "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"
        }
    ]
}
```

출력:

```
{
    "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}\n"
    }
}
```

- 자세한 API 내용은 명령 참조PutImage의 섹션을 참조하세요. AWS CLI

### put-lifecycle-policy

다음 코드 예시에서는 put-lifecycle-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

수명 주기 정책을 생성하려면

다음 put-lifecycle-policy 예제에서는 계정의 기본 레지스트리에 지정된 리포지토리에 대한 수명 주기 정책을 생성합니다.

```
aws ecr put-lifecycle-policy \
    --repository-name "project-a/amazon-ecs-sample" \
    --lifecycle-policy-text "file://policy.json"
```

policy.json의 콘텐츠:

```
{
   "rules": [
       {
           "rulePriority": 1,
           "description": "Expire images older than 14 days",
           "selection": {
               "tagStatus": "untagged",
```

```
                "countType": "sinceImagePushed",
                "countUnit": "days",
                "countNumber": 14
            },
            "action": {
                "type": "expire"
            }
        }
    ]
}
```

출력:

```
{
    "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\"}}]}"
}
```

자세한 내용은 Amazon ECR 사용 설명서의 수명 주기 정책을 참조하세요.

- 자세한 API 내용은 명령 참조PutLifecyclePolicy의 섹션을 참조하세요. AWS CLI

### set-repository-policy

다음 코드 예시에서는 set-repository-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리에 대한 리포지토리 정책을 설정하려면

다음 set-repository-policy 예제에서는 파일에 포함된 리포지토리 정책을 cluster-autoscaler리포지토리에 연결합니다.

```
aws ecr set-repository-policy \
    --repository-name cluster-autoscaler \
    --policy-text file://my-policy.json
```

my-policy.json의 콘텐츠:

```
{
    "Version" : "2008-10-17",
    "Statement" : [
        {
            "Sid" : "allow public pull",
            "Effect" : "Allow",
            "Principal" : "*",
            "Action" : [
                "ecr:BatchCheckLayerAvailability",
                "ecr:BatchGetImage",
                "ecr:GetDownloadUrlForLayer"
            ]
        }
    ]
}
```

출력:

```
{
    "registryId": "012345678910",
    "repositoryName": "cluster-autoscaler",
    "policyText": "{\n  \"Version\" : \"2008-10-17\",\n  \"Statement\" : [ {\n
 \"Sid\" : \"allow public pull\",\n    \"Effect\" : \"Allow\",\n    \"Principal\" :
 \"*\",\n    \"Action\" : [ \"ecr:BatchCheckLayerAvailability\", \"ecr:BatchGetImage
\", \"ecr:GetDownloadUrlForLayer\" ]\n  } ]\n}"
}
```

- 자세한 API 내용은 명령 참조SetRepositoryPolicy의 섹션을 참조하세요. AWS CLI

### start-image-scan

다음 코드 예시에서는 start-image-scan을 사용하는 방법을 보여 줍니다.

AWS CLI

이미지 취약성 스캔을 시작하려면

다음 start-image-scan 예제에서는 지정된 리포지토리의 이미지 다이제스트에 의해 지정된 및에 대한 이미지 스캔을 시작합니다.

```
aws ecr start-image-scan \
```

```
    --repository-name sample-repo \
    --image-
id imageDigest=sha256:74b2c688c700ec95a93e478cdb959737c148df3fbf5ea706abe0318726e885e6
```

출력:

```
{
   "registryId": "012345678910",
   "repositoryName": "sample-repo",
   "imageId": {
       "imageDigest":
 "sha256:74b2c688c700ec95a93e478cdb959737c148df3fbf5ea706abe0318726e885e6"
   },
   "imageScanStatus": {
       "status": "IN_PROGRESS"
   }
}
```

자세한 내용은 Amazon ECR 사용 설명서의 이미지 스캔을 참조하세요.

- 자세한 API 내용은 명령 참조StartImageScan의 섹션을 참조하세요. AWS CLI

### start-lifecycle-policy-preview

다음 코드 예시에서는 start-lifecycle-policy-preview을 사용하는 방법을 보여 줍니다.

AWS CLI

수명 주기 정책 미리 보기를 생성하려면

다음 start-lifecycle-policy-preview 예제에서는 지정된 리포지토리에 대한 JSON 파일로 정의된 수명 주기 정책 미리 보기를 생성합니다.

```
aws ecr start-lifecycle-policy-preview \
    --repository-name "project-a/amazon-ecs-sample" \
    --lifecycle-policy-text "file://policy.json"
```

policy.json의 콘텐츠:

```
{
   "rules": [
       {
```

```
            "rulePriority": 1,
            "description": "Expire images older than 14 days",
            "selection": {
                "tagStatus": "untagged",
                "countType": "sinceImagePushed",
                "countUnit": "days",
                "countNumber": 14
            },
            "action": {
                "type": "expire"
            }
        }
    ]
}
```

출력:

```
{
    "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}\n",
    "status": "IN_PROGRESS"
}
```

- 자세한 API 내용은 명령 참조StartLifecyclePolicyPreview의 섹션을 참조하세요. AWS CLI

### `tag-resource`

다음 코드 예시에서는 `tag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리에 태그를 지정하려면

다음 `tag-resource` 예제에서는 `hello-world`리포지토리`Integ`에 키`Stage`와 값이 있는 태그를 설정합니다.

```
aws ecr tag-resource \
    --resource-arn arn:aws:ecr:us-west-2:012345678910:repository/hello-world \
    --tags Key=Stage,Value=Integ
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### untag-resource

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리포지토리의 태그를 해제하려면

다음 untag-resource 예제에서는 hello-world리포지토리Stage에서 키가 있는 태그를 제거합니다.

```
aws ecr untag-resource \
    --resource-arn arn:aws:ecr:us-west-2:012345678910:repository/hello-world \
    --tag-keys Stage
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### upload-layer-part

다음 코드 예시에서는 upload-layer-part을 사용하는 방법을 보여 줍니다.

AWS CLI

계층 부분을 업로드하려면

다음은 이미지 계층 부분을 layer-test리포지토리에 upload-layer-part 업로드합니다.

```
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
```

출력:

```
{
    "uploadId": "6cb64b8a-9378-0e33-2ab1-b780fab8a9e9",
    "registryId": "012345678910",
    "lastByteReceived": 8323314,
    "repositoryName": "layer-test"
}
```

- 자세한 API 내용은 명령 참조UploadLayerPart의 섹션을 참조하세요. AWS CLI

## 를 사용한 Amazon ECR Public 예제 AWS CLI

다음 코드 예제에서는 Amazon ECR Public과 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `create-repository`

다음 코드 예시에서는 `create-repository`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 퍼블릭 레지스트리에서 리포지토리를 생성하려면

다음 `create-repository` 예제에서는 퍼블릭 레지스트리`project-a/nginx-web-app`에 이름이 지정된 리포지토리를 생성합니다.

```
aws ecr-public create-repository \
    --repository-name project-a/nginx-web-app
```

출력:

```
{
    "repository": {
        "repositoryArn": "arn:aws:ecr-public::123456789012:repository/project-a/
nginx-web-app",
        "registryId": "123456789012",
        "repositoryName": "project-a/nginx-web-app",
        "repositoryUri": "public.ecr.aws/public-registry-custom-alias/project-a/
nginx-web-app",
        "createdAt": "2024-07-01T21:08:55.131000+00:00"
    },
    "catalogData": {}
}
```

자세한 내용은 Amazon 퍼블릭 사용 설명서의 ECR 퍼블릭 리포지토리 생성을 참조하세요.

예 2: 리포지토리의 이미지가 호환되는 리포지토리, 시스템 및 운영 아키텍처의 콘텐츠에 대한 간단한 설명을 사용하여 퍼블릭 레지스트리에서 리포지토리를 생성하는 방법

다음 `create-repository` 예제에서는 리포지토리의 이미지가 호환되는 리포지토리, 시스템 및 운영 아키텍처의 콘텐츠에 대한 간단한 설명을 사용하여 `project-a/nginx-web-app` 퍼블릭 레지스트리에 이름이 지정된 리포지토리를 생성합니다.

```
aws ecr-public create-repository \
    --repository-name project-a/nginx-web-app \
    --catalog-data 'description=My project-a ECR Public
 Repository,architectures=ARM,ARM 64,x86,x86-64,operatingSystems=Linux'
```

출력:

```
{
    "repository": {
        "repositoryArn": "arn:aws:ecr-public::123456789012:repository/project-a/
nginx-web-app",
        "registryId": "123456789012",
        "repositoryName": "project-a/nginx-web-app",
        "repositoryUri": "public.ecr.aws/public-registry-custom-alias/project-a/
nginx-web-app",
```

```
        "createdAt": "2024-07-01T21:23:20.455000+00:00"
    },
    "catalogData": {
        "description": "My project-a ECR Public Repository",
        "architectures": [
            "ARM",
            "ARM 64",
            "x86",
            "x86-64"
        ],
        "operatingSystems": [
            "Linux"
        ]
    }
}
```

자세한 내용은 Amazon 퍼블릭 사용 설명서의 ECR 퍼블릭 리포지토리 생성을 참조하세요.

예제 3: , logoImageBlobaboutText, usageText 태그 정보와 함께 퍼블릭 레지스트리에 리포지토리를 생성하려면

다음 `create-repository` 예제에서는 logoImageBlob, aboutText, usageText 태그 정보와 함께 퍼블릭 레지스트리nginx-web-app 에 project-a/라는 리포지토리를 생성합니다.

```
aws ecr-public create-repository \
    --cli-input-json file://myfile.json
```

myfile.json의 콘텐츠:

```
{
    "repositoryName": "project-a/nginx-web-app",
    "catalogData": {
        "description": "My project-a ECR Public Repository",
        "architectures": [
            "ARM",
            "ARM 64",
            "x86",
            "x86-64"
        ],
        "operatingSystems": [
            "Linux"
        ],
        "logoImageBlob": "iVBORw0KGgoA<<truncated-for-better-reading>>ErkJggg==",
```

```
        "aboutText": "## Quick reference\n\nMaintained by: [the Amazon Linux Team]
(https://github.com/aws/amazon-linux-docker-images)\n\nWhere to get help: [the
 Docker Community Forums](https://forums.docker.com/), [the Docker Community Slack]
(https://dockr.ly/slack), or [Stack Overflow](https://stackoverflow.com/search?
tab=newest&q=docker)\n\n## Supported tags and respective `dockerfile` links\n\n*
 [`2.0.20200722.0`, `2`, `latest`](https://github.com/amazonlinux/container-images/
blob/03d54f8c4d522bf712cffd6c8f9aafba0a875e78/Dockerfile)\n* [`2.0.20200722.0-
with-sources`, `2-with-sources`, `with-sources`](https://github.com/
amazonlinux/container-images/blob/1e7349845e029a2e6afe6dc473ef17d052e3546f/
Dockerfile)\n* [`2018.03.0.20200602.1`, `2018.03`, `1`](https://github.com/
amazonlinux/container-images/blob/f10932e08c75457eeb372bf1cc47ea2a4b8e98c8/
Dockerfile)\n* [`2018.03.0.20200602.1-with-sources`, `2018.03-with-sources`,
 `1-with-sources`](https://github.com/amazonlinux/container-images/
blob/8c9ee491689d901aa72719be0ec12087a5fa8faf/Dockerfile)\n\n## What is Amazon
 Linux?\n\nAmazon Linux is provided by Amazon Web Services (AWS). It is designed
 to provide a stable, secure, and high-performance execution environment for
 applications running on Amazon EC2. The full distribution includes packages that
 enable easy integration with AWS, including launch configuration tools and many
 popular AWS libraries and tools. AWS provides ongoing security and maintenance
 updates to all instances running Amazon Linux.\n\nThe Amazon Linux container image
 contains a minimal set of packages. To install additional packages, [use `yum`]
(https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/managing-software.html).\n\nAWS
 provides two versions of Amazon Linux: [Amazon Linux 2](https://aws.amazon.com/
amazon-linux-2/) and [Amazon Linux AMI](https://aws.amazon.com/amazon-linux-ami/).
\n\nFor information on security updates for Amazon Linux, please refer to [Amazon
 Linux 2 Security Advisories](https://alas.aws.amazon.com/alas2.html) and [Amazon
 Linux AMI Security Advisories](https://alas.aws.amazon.com/). Note that Docker
 Hub's vulnerability scanning for Amazon Linux is currently based on RPM versions,
 which does not reflect the state of backported patches for vulnerabilities.\n
\n## Where can I run Amazon Linux container images?\n\nYou can run Amazon Linux
 container images in any Docker based environment. Examples include, your laptop,
 in Amazon EC2 instances, and Amazon ECS clusters.\n\n## License\n\nAmazon Linux is
 available under the [GNU General Public License, version 2.0](https://github.com/
aws/amazon-linux-docker-images/blob/master/LICENSE). Individual software packages
 are available under their own licenses; run `rpm -qi [package name]` or check
 `/usr/share/doc/[package name]-*` and `/usr/share/licenses/[package name]-*` for
 details.\n\nAs with all Docker images, these likely also contain other software
 which may be under other licenses (such as Bash, etc from the base distribution,
 along with any direct or indirect dependencies of the primary software being
 contained).\n\nSome additional license information which was able to be auto-
detected might be found in [the `repo-info` repository's `amazonlinux/` directory]
(https://github.com/docker-library/repo-info/tree/master/repos/amazonlinux).\n\n##
 Security\n\nFor information on security updates for Amazon Linux, please refer
 to [Amazon Linux 2 Security Advisories](https://alas.aws.amazon.com/alas2.html)
```

```
 and [Amazon Linux AMI Security Advisories](https://alas.aws.amazon.com/). Note
 that Docker Hub's vulnerability scanning for Amazon Linux is currently based
 on RPM versions, which does not reflect the state of backported patches for
 vulnerabilities.",
        "usageText": "## Supported architectures\n\namd64, arm64v8\n\n## Where
 can I run Amazon Linux container images?\n\nYou can run Amazon Linux container
 images in any Docker based environment. Examples include, your laptop, in Amazon
 EC2 instances, and ECS clusters.\n\n## How do I install a software package from
 Extras repository in Amazon Linux 2?\n\nAvailable packages can be listed with the
 `amazon-linux-extras` command. Packages can be installed with the `amazon-linux-
extras install <package>` command. Example: `amazon-linux-extras install rust1`\n
\n## Will updates be available for Amazon Linux containers?\n\nSimilar to the Amazon
 Linux images for Amazon EC2 and on-premises use, Amazon Linux container images will
 get ongoing updates from Amazon in the form of security updates, bug fix updates,
 and other enhancements. Security bulletins for Amazon Linux are available at
 https://alas.aws.amazon.com/\n\n## Will AWS Support the current version of Amazon
 Linux going forward?\n\nYes; in order to avoid any disruption to your existing
 applications and to facilitate migration to Amazon Linux 2, AWS will provide
 regular security updates for Amazon Linux 2018.03 AMI and container image for 2
 years after the final LTS build is announced. You can also use all your existing
 support channels such as AWS Support and Amazon Linux Discussion Forum to continue
 to submit support requests."
    },
    "tags": [
        {
            "Key": "Name",
            "Value": "project-a/nginx-web-app"
        },
        {
            "Key": "Environment",
            "Value": "Prod"
        }
    ]
}
```

출력:

```
{
    "repository": {
        "repositoryArn": "arn:aws:ecr-public::123456789012:repository/project-a/
nginx-web-app",
        "registryId": "123456789012",
        "repositoryName": "project-a/nginx-web-app",
```

```
        "repositoryUri": "public.ecr.aws/public-registry-custom-alias/project-a/
nginx-web-app",
        "createdAt": "2024-07-01T21:53:05.749000+00:00"
    },
    "catalogData": {
        "description": "My project-a ECR Public Repository",
        "architectures": [
            "ARM",
            "ARM 64",
            "x86",
            "x86-64"
        ],
        "operatingSystems": [
            "Linux"
        ],
        "logoUrl": "https://d3g9o9u8re44ak.cloudfront.net/
logo/23861450-4b9b-403c-9a4c-7aa0ef140bb8/2f9bf5a7-a32f-45b4-b5cd-c5770a35e6d7.png",
        "aboutText": "## Quick reference\n\nMaintained by: [the Amazon Linux Team]
(https://github.com/aws/amazon-linux-docker-images)\n\nWhere to get help: [the
 Docker Community Forums](https://forums.docker.com/), [the Docker Community Slack]
(https://dockr.ly/slack), or [Stack Overflow](https://stackoverflow.com/search?
tab=newest&q=docker)\n\n## Supported tags and respective `dockerfile` links\n\n*
 [`2.0.20200722.0`, `2`, `latest`](https://github.com/amazonlinux/container-images/
blob/03d54f8c4d522bf712cffd6c8f9aafba0a875e78/Dockerfile)\n* [`2.0.20200722.0-
with-sources`, `2-with-sources`, `with-sources`](https://github.com/
amazonlinux/container-images/blob/1e7349845e029a2e6afe6dc473ef17d052e3546f/
Dockerfile)\n* [`2018.03.0.20200602.1`, `2018.03`, `1`](https://github.com/
amazonlinux/container-images/blob/f10932e08c75457eeb372bf1cc47ea2a4b8e98c8/
Dockerfile)\n* [`2018.03.0.20200602.1-with-sources`, `2018.03-with-sources`,
 `1-with-sources`](https://github.com/amazonlinux/container-images/
blob/8c9ee491689d901aa72719be0ec12087a5fa8faf/Dockerfile)\n\n## What is Amazon
 Linux?\n\nAmazon Linux is provided by Amazon Web Services (AWS). It is designed
 to provide a stable, secure, and high-performance execution environment for
 applications running on Amazon EC2. The full distribution includes packages that
 enable easy integration with AWS, including launch configuration tools and many
 popular AWS libraries and tools. AWS provides ongoing security and maintenance
 updates to all instances running Amazon Linux.\n\nThe Amazon Linux container image
 contains a minimal set of packages. To install additional packages, [use `yum`]
(https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/managing-software.html).\n\nAWS
 provides two versions of Amazon Linux: [Amazon Linux 2](https://aws.amazon.com/
amazon-linux-2/) and [Amazon Linux AMI](https://aws.amazon.com/amazon-linux-ami/).
\n\nFor information on security updates for Amazon Linux, please refer to [Amazon
 Linux 2 Security Advisories](https://alas.aws.amazon.com/alas2.html) and [Amazon
 Linux AMI Security Advisories](https://alas.aws.amazon.com/). Note that Docker
```

```
 Hub's vulnerability scanning for Amazon Linux is currently based on RPM versions,
 which does not reflect the state of backported patches for vulnerabilities.\n
\n## Where can I run Amazon Linux container images?\n\nYou can run Amazon Linux
 container images in any Docker based environment. Examples include, your laptop,
 in Amazon EC2 instances, and Amazon ECS clusters.\n\n## License\n\nAmazon Linux is
 available under the [GNU General Public License, version 2.0](https://github.com/
aws/amazon-linux-docker-images/blob/master/LICENSE). Individual software packages
 are available under their own licenses; run `rpm -qi [package name]` or check
 `/usr/share/doc/[package name]-*` and `/usr/share/licenses/[package name]-*` for
 details.\n\nAs with all Docker images, these likely also contain other software
 which may be under other licenses (such as Bash, etc from the base distribution,
 along with any direct or indirect dependencies of the primary software being
 contained).\n\nSome additional license information which was able to be auto-
detected might be found in [the `repo-info` repository's `amazonlinux/` directory]
(https://github.com/docker-library/repo-info/tree/master/repos/amazonlinux).\n\n##
 Security\n\nFor information on security updates for Amazon Linux, please refer
 to [Amazon Linux 2 Security Advisories](https://alas.aws.amazon.com/alas2.html)
 and [Amazon Linux AMI Security Advisories](https://alas.aws.amazon.com/). Note
 that Docker Hub's vulnerability scanning for Amazon Linux is currently based
 on RPM versions, which does not reflect the state of backported patches for
 vulnerabilities.",
       "usageText": "## Supported architectures\n\namd64, arm64v8\n\n## Where
 can I run Amazon Linux container images?\n\nYou can run Amazon Linux container
 images in any Docker based environment. Examples include, your laptop, in Amazon
 EC2 instances, and ECS clusters.\n\n## How do I install a software package from
 Extras repository in Amazon Linux 2?\n\nAvailable packages can be listed with the
 `amazon-linux-extras` command. Packages can be installed with the `amazon-linux-
extras install <package>` command. Example: `amazon-linux-extras install rust1`\n
\n## Will updates be available for Amazon Linux containers?\n\nSimilar to the Amazon
 Linux images for Amazon EC2 and on-premises use, Amazon Linux container images will
 get ongoing updates from Amazon in the form of security updates, bug fix updates,
 and other enhancements. Security bulletins for Amazon Linux are available at
 https://alas.aws.amazon.com/\n\n## Will AWS Support the current version of Amazon
 Linux going forward?\n\nYes; in order to avoid any disruption to your existing
 applications and to facilitate migration to Amazon Linux 2, AWS will provide
 regular security updates for Amazon Linux 2018.03 AMI and container image for 2
 years after the final LTS build is announced. You can also use all your existing
 support channels such as AWS Support and Amazon Linux Discussion Forum to continue
 to submit support requests."
    }
}
```

자세한 내용은 Amazon Public 사용 설명서의 퍼블릭 리포지토리 생성 및 Amazon Public 사용 설명서의 리포지토리 카탈로그 데이터를 참조하세요. ECR ECR

- 자세한 API 내용은 명령 참조CreateRepository의 섹션을 참조하세요. AWS CLI

### `delete-repository`

다음 코드 예시에서는 `delete-repository`을 사용하는 방법을 보여 줍니다.

AWS CLI

퍼블릭 레지스트리에서 리포지토리를 삭제하려면

다음 `delete-repository` 예제에서는 퍼블릭 레지스트리`project-a/nginx-web-app`에서 이름이 지정된 리포지토리를 삭제합니다.

```
aws ecr-public delete-repository \
    --repository-name project-a/nginx-web-app
```

출력:

```
{
    "repository": {
        "repositoryArn": "arn:aws:ecr-public::123456789012:repository/project-a/
nginx-web-app",
        "registryId": "123456789012",
        "repositoryName": "project-a/nginx-web-app",
        "repositoryUri": "public.ecr.aws/public-registry-custom-alias/project-a/
nginx-web-app",
        "createdAt": "2024-07-01T22:14:50.103000+00:00"
    }
}
```

자세한 내용은 Amazon 퍼블릭 사용 설명서의 ECR 퍼블릭 리포지토리 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteRepository의 섹션을 참조하세요. AWS CLI

## 를 사용한 Amazon ECS 예제 AWS CLI

다음 코드 예제에서는 Amazon 에서 를 사용하여 작업을 수행하고 일반적인 시나리오 AWS Command Line Interface 를 구현하는 방법을 보여줍니다ECS.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `create-capacity-provider`

다음 코드 예시에서는 `create-capacity-provider`을 사용하는 방법을 보여 줍니다.

AWS CLI

용량 공급자를 생성하려면

다음 create-capacity-provider 예제에서는 내 라는 Auto Scaling 그룹을 사용하고 관리형 스케일링 및 관리형 종료 보호ASG가 활성화된 용량 공급자를 생성합니다. 이 구성은 Amazon ECS 클러스터 자동 조정에 사용됩니다.

```
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=ENABLE
```

출력:

```
{
    "capacityProvider": {
    "capacityProviderArn": "arn:aws:ecs:us-east-1:123456789012:capacity-provider/
MyCapacityProvider",
    "name": "MyCapacityProvider",
    "status": "ACTIVE",
    "autoScalingGroupProvider": {
```

```
        "autoScalingGroupArn": "arn:aws:autoscaling:us-
east-1:132456789012:autoScalingGroup:57ffcb94-11f0-4d6d-
bf60-3bac5EXAMPLE:autoScalingGroupName/MyASG",
        "managedScaling": {
            "status": "ENABLED",
            "targetCapacity": 100,
            "minimumScalingStepSize": 1,
            "maximumScalingStepSize": 10000,
            "instanceWarmupPeriod": 300
        },
        "managedTerminationProtection": "ENABLED"
    },
    "tags": []
}
```

자세한 내용은 Amazon 개발자 안내서의 Amazon ECS 클러스터 자동 조정을 참조하세요. ECS

- 자세한 API 내용은 명령 참조CreateCapacityProvider의 섹션을 참조하세요. AWS CLI

### create-cluster

다음 코드 예시에서는 create-cluster을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: 새 클러스터를 생성하는 방법

다음 create-cluster 예시에서는 클러스터를 생성합니다.

```
aws ecs create-cluster \
    --cluster-name MyCluster
```

출력:

```
{
    "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": []
    }
}
```

자세한 내용은 Amazon ECS 개발자 안내서의 클러스터 생성을 참조하세요.

예 2: 용량 공급자를 사용하여 새 클러스터를 생성하는 방법

다음 `create-cluster` 예시에서는 클러스터를 생성하고 기존 용량 공급자 2개를 클러스터에 연결합니다. `create-capacity-provider` 명령을 사용하여 용량 공급자를 생성합니다. 기본 용량 공급자 전략을 지정하는 것은 선택 사항이지만 권장됩니다. 이 예시에서는 이름이 `MyCluster`인 클러스터를 생성하고 여기에 `MyCapacityProvider1` 및 `MyCapacityProvider2` 용량 공급자를 연결합니다. 기본 용량 공급자 전략이 지정되어 태스크를 두 용량 공급자 모두에 균등하게 분산합니다.

aws ecs create-cluster --cluster-name MyCluster --capacity-providers MyCapacityProvider1 MyCapacityProvider2 --default-capacity-provider-strategy capacityProvider=MyCapacityProvider1,weight=1 capacityProvider=MyCapacityProvider2,weight=1

출력:

```
{
    "cluster": {
        "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",
        "clusterName": "MyCluster",
        "status": "PROVISIONING",
        "registeredContainerInstancesCount": 0,
        "pendingTasksCount": 0,
        "runningTasksCount": 0,
        "activeServicesCount": 0,
        "statistics": [],
        "settings": [
            {
                "name": "containerInsights",
                "value": "enabled"
            }
        ],
        "capacityProviders": [
            "MyCapacityProvider1",
            "MyCapacityProvider2"
```

```
        ],
        "defaultCapacityProviderStrategy": [
            {
                "capacityProvider": "MyCapacityProvider1",
                "weight": 1,
                "base": 0
            },
            {
                "capacityProvider": "MyCapacityProvider2",
                "weight": 1,
                "base": 0
            }
        ],
        "attachments": [
           {
                "id": "0fb0c8f4-6edd-4de1-9b09-17e470ee1918",
                "type": "asp",
                "status": "PRECREATED",
                "details": [
                    {
                        "name": "capacityProviderName",
                        "value": "MyCapacityProvider1"
                    },
                    {
                        "name": "scalingPlanName",
                        "value": "ECSManagedAutoScalingPlan-a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111"
                    }
                ]
            },
            {
                "id": "ae592060-2382-4663-9476-b015c685593c",
                "type": "asp",
                "status": "PRECREATED",
                "details": [
                    {
                        "name": "capacityProviderName",
                        "value": "MyCapacityProvider2"
                    },
                    {
                        "name": "scalingPlanName",
                        "value": "ECSManagedAutoScalingPlan-a1b2c3d4-5678-90ab-cdef-
EXAMPLE22222"
                    }
```

```
                ]
            }
        ],
        "attachmentsStatus": "UPDATE_IN_PROGRESS"
    }
}
```

자세한 내용은 Amazon ECS 개발자 안내서의 클러스터 용량 공급자를 참조하세요.

예 3: 여러 태그가 포함된 새 클러스터를 생성하는 방법

다음 `create-cluster` 예시에서는 여러 태그가 있는 클러스터를 만듭니다. 단축형 구문을 사용하여 태그를 추가하는 방법에 대한 자세한 내용은 AWS CLI 사용 설명서의 AWS 명령줄 인터페이스에서 단축형 구문 사용을 참조하세요.

```
aws ecs create-cluster \
    --cluster-name MyCluster \
    --tags key=key1,value=value1 key=key2,value=value2 key=key3,value=value3
```

출력:

```
{
    "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"
```

```
            }
        ]
    }
}
```

자세한 내용은 Amazon ECS 개발자 안내서의 클러스터 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateCluster의 섹션을 참조하세요. AWS CLI

### `create-service`

다음 코드 예시에서는 `create-service`을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: Fargate 태스크를 사용하여 서비스를 생성하는 방법

다음 `create-service` 예시에서는 Fargate 태스크를 사용하여 서비스를 생성하는 방법을 보여줍니다.

```
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
```

출력:

```
{
    "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"
            }
        ],
        "enableECSManagedTags": false,
        "propagateTags": "NONE"
    }
}
```

예제 2: EC2 시작 유형을 사용하여 서비스를 생성하려면

다음 `create-service` 예제에서는 EC2 시작 유형을 사용하는 작업`ecs-simple-service`으로를 호출하는 서비스를 생성하는 방법을 보여줍니다. 이 서비스는 `sleep360` 태스크 정의를 사용하며 태스크의 인스턴스화 1개를 유지 관리합니다.

```
aws ecs create-service \
    --cluster MyCluster \
    --service-name ecs-simple-service \
    --task-definition sleep360:2 \
    --desired-count 1
```

출력:

```
{
    "service": {
        "serviceArn": "arn:aws:ecs:us-west-2:123456789012:service/MyCluster/ecs-
simple-service",
        "serviceName": "ecs-simple-service",
        "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",
        "loadBalancers": [],
        "serviceRegistries": [],
        "status": "ACTIVE",
        "desiredCount": 1,
        "runningCount": 0,
        "pendingCount": 0,
        "launchType": "EC2",
        "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/
sleep360:2",
        "deploymentConfiguration": {
            "maximumPercent": 200,
            "minimumHealthyPercent": 100
        },
        "deployments": [
            {
                "id": "ecs-svc/1234567890123456789",
                "status": "PRIMARY",
                "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-
definition/sleep360:2",
                "desiredCount": 1,
                "pendingCount": 0,
                "runningCount": 0,
                "createdAt": 1557206498.798,
                "updatedAt": 1557206498.798,
                "launchType": "EC2"
            }
        ],
        "events": [],
        "createdAt": 1557206498.798,
        "placementConstraints": [],
        "placementStrategy": [],
        "schedulingStrategy": "REPLICA",
        "enableECSManagedTags": false,
        "propagateTags": "NONE"
    }
}
```

예 3: 외부 배포 컨트롤러를 사용하는 서비스를 생성하는 방법

다음 create-service 예시에서는 외부 배포 컨트롤러를 사용하는 서비스를 생성합니다.

```
aws ecs create-service \
    --cluster MyCluster \
    --service-name MyService \
    --deployment-controller type=EXTERNAL \
    --desired-count 1
```

출력:

```
{
    "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"
```

```
    }
}
```

예 4: 로드 밸런서 뒤에 새 서비스를 생성하는 방법

다음 `create-service` 예시에서는 로드 밸런서 뒤에 있는 서비스를 생성하는 방법을 보여줍니다. 컨테이너 인스턴스와 동일한 리전에 로드 밸런서가 구성되어 있어야 합니다. 이 예제에서는 다음 콘텐츠와 `ecs-simple-service-elb.json` 함께 라는 `--cli-input-json` 옵션과 JSON 입력 파일을 사용합니다.

```
{
    "serviceName": "ecs-simple-service-elb",
    "taskDefinition": "ecs-demo",
    "loadBalancers": [
        {
            "loadBalancerName": "EC2Contai-EcsElast-123456789012",
            "containerName": "simple-demo",
            "containerPort": 80
        }
    ],
    "desiredCount": 10,
    "role": "ecsServiceRole"
}
```

명령:

```
aws ecs create-service \
    --cluster MyCluster \
    --service-name ecs-simple-service-elb \
    --cli-input-json file://ecs-simple-service-elb.json
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 Amazon ECS 개발자 안내서의 서비스 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateService의 섹션을 참조하세요. AWS CLI

### `create-task-set`

다음 코드 예시에서는 `create-task-set`을 사용하는 방법을 보여 줍니다.

AWS CLI

작업 세트를 생성하려면

다음 `create-task-set` 예제에서는 외부 배포 컨트롤러를 사용하는 서비스에서 태스크 세트를 생성합니다.

```
aws ecs create-task-set \
    --cluster MyCluster \
    --service MyService \
    --task-definition MyTaskDefinition:2 \
    --network-
configuration "awsvpcConfiguration={subnets=[subnet-12344321],securityGroups=[sg-12344321]}"
```

출력:

```
{
    "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
    }
```

```
}
```

- 자세한 API 내용은 명령 참조CreateTaskSet의 섹션을 참조하세요. AWS CLI

### `delete-account-setting`

다음 코드 예시에서는 `delete-account-setting`을 사용하는 방법을 보여 줍니다.

AWS CLI

특정 IAM 사용자 또는 IAM 역할에 대한 계정 설정을 삭제하려면

다음 예제에서는 특정 IAM 사용자 또는 IAM 역할에 대한 계정 설정을 `delete-account-setting` 삭제합니다.

```
aws ecs delete-account-setting \
    --name serviceLongArnFormat \
    --principal-arn arn:aws:iam::123456789012:user/MyUser
```

출력:

```
{
    "setting": {
        "name": "serviceLongArnFormat",
        "value": "enabled",
        "principalArn": "arn:aws:iam::123456789012:user/MyUser"
    }
}
```

자세한 내용은 Amazon 개발자 안내서의 Amazon 리소스 이름(ARNs) 및 IDs 섹션을 참조하세요. ECS

- 자세한 API 내용은 명령 참조DeleteAccountSetting의 섹션을 참조하세요. AWS CLI

### `delete-attributes`

다음 코드 예시에서는 `delete-attributes`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon ECS 리소스에서 사용자 지정 속성을 하나 이상 삭제하려면

다음은 컨테이너 인스턴스stack에서 이름이 인 속성을 delete-attributes 삭제합니다.

```
aws ecs delete-attributes \
    --attributes name=stack,targetId=arn:aws:ecs:us-west-2:130757420319:container-instance/1c3be8ed-df30-47b4-8f1e-6e68ebd01f34
```

출력:

```
{
    "attributes": [
        {
            "name": "stack",
            "targetId": "arn:aws:ecs:us-west-2:130757420319:container-instance/1c3be8ed-df30-47b4-8f1e-6e68ebd01f34",
            "value": "production"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DeleteAttributes의 섹션을 참조하세요. AWS CLI

### delete-capacity-provider

다음 코드 예시에서는 delete-capacity-provider을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: Amazon 리소스 이름(ARN)을 사용하여 용량 공급자를 삭제하려면

다음 delete-capacity-provider 예제에서는 용량 공급자의 Amazon 리소스 이름(ARN)을 지정하여 용량 공급자를 삭제합니다. ARN 와 용량 공급자 삭제 상태는 describe-capacity-providers 명령을 사용하여 검색할 수 있습니다.

```
aws ecs delete-capacity-provider \
    --capacity-provider arn:aws:ecs:us-west-2:123456789012:capacity-provider/ExampleCapacityProvider
```

출력:

```
{
```

```
    "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": []
    }
}
```

자세한 내용은 Amazon ECS 개발자 안내서의 클러스터 용량 공급자를 참조하세요.

예제 2: 이름을 사용하여 용량 공급자를 삭제하려면

다음 `delete-capacity-provider` 예제에서는 용량 공급자의 짧은 이름을 지정하여 용량 공급자를 삭제합니다. 짧은 이름과 용량 공급자 삭제 상태는 `describe-capacity-providers` 명령을 사용하여 검색할 수 있습니다.

```
aws ecs delete-capacity-provider \
    --capacity-provider ExampleCapacityProvider
```

출력:

```
{
    "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": []
    }
}
```

자세한 내용은 Amazon ECS 개발자 안내서의 클러스터 용량 공급자를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteCapacityProvider의 섹션을 참조하세요. AWS CLI

### delete-cluster

다음 코드 예시에서는 delete-cluster을 사용하는 방법을 보여 줍니다.

AWS CLI

빈 클러스터를 삭제하는 방법

다음 delete-cluster 예시에서는 지정된 빈 클러스터를 삭제합니다.

```
aws ecs delete-cluster --cluster MyCluster
```

출력:

```
{
    "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": []
    }
}
```

자세한 내용은 Amazon ECS 개발자 안내서의 클러스터 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteCluster의 섹션을 참조하세요. AWS CLI

### delete-service

다음 코드 예시에서는 delete-service을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스를 삭제하는 방법

다음 ecs delete-service 예시에서는 클러스터에서 지정된 서비스를 삭제합니다. --force 파라미터를 포함하면 태스크가 없도록 축소되지 않은 서비스도 삭제할 수 있습니다.

```
aws ecs delete-service --cluster MyCluster --service MyService1 --force
```

자세한 내용은 Amazon ECS 개발자 안내서의 서비스 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteService의 섹션을 참조하세요. AWS CLI

### delete-task-definitions

다음 코드 예시에서는 delete-task-definitions을 사용하는 방법을 보여 줍니다.

AWS CLI

작업 정의를 삭제하려면

다음 delete-task-definitions 예제에서는 INACTIVE 작업 정의를 삭제합니다.

```
aws ecs delete-task-definitions \
    --task-definition curltest:1
```

출력:

```
{
"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": []
}
```

자세한 내용은 Amazon 개발자 안내서의 Amazon ECS 작업 정의를 참조하세요. ECS

- 자세한 API 내용은 명령 참조DeleteTaskDefinitions의 섹션을 참조하세요. AWS CLI

### delete-task-set

다음 코드 예시에서는 delete-task-set을 사용하는 방법을 보여 줍니다.

AWS CLI

작업 세트를 삭제하려면

다음 delete-task-set 예제에서는 작업 세트를 삭제하는 방법을 보여줍니다. 0으로 조정되지 않았더라도 파라미터를 포함하여 태스크 세트를 --force 삭제할 수 있습니다.

```
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
```

출력:

```
{
    "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
    }
}
```

- 자세한 API 내용은 명령 참조DeleteTaskSet의 섹션을 참조하세요. AWS CLI

### deregister-container-instance

다음 코드 예시에서는 deregister-container-instance을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터에서 컨테이너 인스턴스 등록을 취소하려면

다음 deregister-container-instance 예제에서는 지정된 클러스터에서 컨테이너 인스턴스의 등록을 취소합니다. 컨테이너 인스턴스에서 여전히 실행 중인 태스크가 있는 경우 등록을 취소하기 전에 해당 태스크를 중지하거나 --force 옵션을 사용해야 합니다.

```
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
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 ECS 개발자 안내서의 컨테이너 인스턴스 등록 취소를 참조하세요.

- 자세한 API 내용은 명령 참조DeregisterContainerInstance의 섹션을 참조하세요. AWS CLI

### deregister-task-definition

다음 코드 예시에서는 deregister-task-definition을 사용하는 방법을 보여 줍니다.

AWS CLI

작업 정의 등록을 취소하려면

다음 deregister-task-definition 예제에서는 기본 리전에서 curler 태스크 정의의 첫 번째 개정을 등록 취소합니다.

```
aws ecs deregister-task-definition --task-definition curler:1
```

결과 출력에서 작업 정의 상태는 를 표시합니다INACTIVE.

```
{
    "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
    }
}
```

자세한 내용은 Amazon 개발자 안내서의 Amazon ECS 작업 정의를 참조하세요. ECS

- 자세한 API 내용은 명령 참조DeregisterTaskDefinition의 섹션을 참조하세요. AWS CLI

### describe-capacity-providers

다음 코드 예시에서는 describe-capacity-providers을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 모든 용량 공급자 설명

다음 describe-capacity-providers 예제에서는 모든 용량 공급자에 대한 세부 정보를 검색합니다.

```
aws ecs describe-capacity-providers
```

출력:

```
{
    "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": []
        }
    ]
}
```

자세한 내용은 Amazon ECS 개발자 안내서의 클러스터 용량 공급자를 참조하세요.

예제 2: 특정 용량 공급자 설명

다음 `describe-capacity-providers` 예제에서는 특정 용량 공급자에 대한 세부 정보를 검색합니다. `--include TAGS` 파라미터를 사용하면 용량 공급자와 연결된 태그가 출력에 추가됩니다.

```
aws ecs describe-capacity-providers \
    --capacity-providers MyCapacityProvider \
    --include TAGS
```

출력:

```
{
    "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"
                }
            ]
        }
    ]
}
```

자세한 내용은 Amazon ECS 개발자 안내서의 클러스터 용량 공급자를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeCapacityProviders의 섹션을 참조하세요. AWS CLI

### describe-clusters

다음 코드 예시에서는 describe-clusters을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: 클러스터를 설명하는 방법

다음 describe-clusters 예시에서는 지정된 클러스터에 대한 세부 정보를 검색합니다.

```
aws ecs describe-clusters \
    --cluster default
```

출력:

```
{
    "clusters": [
        {
            "status": "ACTIVE",
            "clusterName": "default",
            "registeredContainerInstancesCount": 0,
            "pendingTasksCount": 0,
            "runningTasksCount": 0,
            "activeServicesCount": 1,
            "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/default"
        }
```

```
    ],
    "failures": []
}
```

자세한 내용은 Amazon 개발자 안내서의 Amazon ECS 클러스터를 참조하세요.  ECS

예 2: 첨부 파일 옵션을 사용하여 클러스터를 설명하는 방법

다음 `describe-clusters` 예제에서는 ATTACHMENTS 옵션을 지정합니다. 지정된 클러스터에 대한 세부 정보와 클러스터에 연결된 리소스 목록을 첨부 파일 형식으로 검색합니다. 클러스터와 함께 용량 공급자를 사용하는 경우 AutoScaling 계획 또는 조정 정책인 리소스는 asp 또는 as_policy 로 표시됩니다ATTACHMENTS.

```
aws ecs describe-clusters \
    --include ATTACHMENTS \
    --clusters sampleCluster
```

출력:

```
{
    "clusters": [
        {
            "clusterArn": "arn:aws:ecs:af-south-1:123456789222:cluster/
sampleCluster",
            "clusterName": "sampleCluster",
            "status": "ACTIVE",
            "registeredContainerInstancesCount": 0,
            "runningTasksCount": 0,
            "pendingTasksCount": 0,
            "activeServicesCount": 0,
            "statistics": [],
            "tags": [],
            "settings": [],
            "capacityProviders": [
                "sampleCapacityProvider"
            ],
            "defaultCapacityProviderStrategy": [],
            "attachments": [
                {
                    "id": "a1b2c3d4-5678-901b-cdef-EXAMPLE22222",
                    "type": "as_policy",
                    "status": "CREATED",
                    "details": [
```

```
                        {
                            "name": "capacityProviderName",
                            "value": "sampleCapacityProvider"
                        },
                        {
                            "name": "scalingPolicyName",
                            "value": "ECSManagedAutoScalingPolicy-3048e262-
fe39-4eaf-826d-6f975d303188"
                        }
                    ]
                }
            ],
            "attachmentsStatus": "UPDATE_COMPLETE"
        }
    ],
    "failures": []
}
```

자세한 내용은 Amazon 개발자 안내서의 Amazon ECS 클러스터를 참조하세요. ECS

- 자세한 API 내용은 명령 참조DescribeClusters의 섹션을 참조하세요. AWS CLI

### describe-container-instances

다음 코드 예시에서는 describe-container-instances을 사용하는 방법을 보여 줍니다.

AWS CLI

컨테이너 인스턴스를 설명하려면

다음 describe-container-instances 예제에서는 컨테이너 인스턴스를 식별자UUID로 사용하여 update 클러스터의 컨테이너 인스턴스에 대한 세부 정보를 검색합니다.

```
aws ecs describe-container-instances \
    --cluster update \
    --container-instances a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

출력:

```
{
    "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"
            }
        }
    ]
}
```

자세한 내용은 Amazon 개발자 안내서의 Amazon ECS 컨테이너 인스턴스를 참조하세요. ECS

- 자세한 API 내용은 명령 참조DescribeContainerInstances의 섹션을 참조하세요. AWS CLI

### describe-services

다음 코드 예시에서는 describe-services을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스를 설명하려면

다음 describe-services 예제에서는 기본 클러스터의 my-http-service 서비스에 대한 세부 정보를 검색합니다.

```
aws ecs describe-services --services my-http-service
```

출력:

```
{
    "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": []
}
```

자세한 내용은 Amazon ECS 개발자 안내서의 서비스를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeServices의 섹션을 참조하세요. AWS CLI

### describe-task-definition

다음 코드 예시에서는 describe-task-definition을 사용하는 방법을 보여 줍니다.

AWS CLI

작업 정의를 설명하려면

다음 describe-task-definition 예제에서는 작업 정의의 세부 정보를 검색합니다.

```
aws ecs describe-task-definition \
    --task-definition hello_world:8
```

출력:

```
{
    "taskDefinition": {
        "taskDefinitionArn": "arn:aws:ecs:us-east-1:012345678910:task-definition/
hello_world:8",
        "containerDefinitions": [
            {
                "cpu": 10,
                "environment": [],
                "essential": true,
                "image": "wordpress",
                "links": [
                    "mysql"
                ] ,
                "memory": 500,
                "mountPoints": [],
                "name": "wordpress",
                "portMappings": [
                    {
```

```
                    "containerPort": 80,
                    "hostPort": 80
                }
            ],
            "volumesFrom": []
        },
        {
            "cpu": 10,
            "environment": [
                {
                    "name": "MYSQL_ROOT_PASSWORD",
                    "value": "password"
                }
            ],
            "essential": true,
            "image": "mysql",
            "memory": 500,
            "mountPoints": [],
            "name": "mysql",
            "portMappings": [],
            "volumesFrom": []
        }
    ],
    "family": "hello_world",
    "revision": 8,
    "volumes": [],
    "status": "ACTIVE",
    "placementConstraints": [],
    "compatibilities": [
        "EXTERNAL",
        "EC2"
    ],
    "registeredAt": "2024-06-21T11:15:12.669000-05:00",
    "registeredBy": "arn:aws:sts::012345678910:assumed-role/demo-role/jane-doe"
    },
    "tags": []
}
```

자세한 내용은 Amazon 개발자 안내서의 Amazon ECS 작업 정의를 참조하세요. ECS

- 자세한 API 내용은 명령 참조DescribeTaskDefinition의 섹션을 참조하세요. AWS CLI

### describe-task-sets

다음 코드 예시에서는 describe-task-sets을 사용하는 방법을 보여 줍니다.

AWS CLI

작업 세트를 설명하려면

다음 describe-task-sets 예제에서는 외부 배포자를 사용하는 서비스의 태스크 세트를 설명합니다.

```
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
```

출력:

```
{
    "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": []
}
```

- 자세한 API 내용은 명령 참조DescribeTaskSets의 섹션을 참조하세요. AWS CLI

### `describe-tasks`

다음 코드 예시에서는 `describe-tasks`을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: 단일 태스크를 설명하는 방법

다음 `describe-tasks` 예시에서는 클러스터의 태스크 세부 정보를 검색합니다. ID 또는 전체 태스크를 사용하여 태스크를 지정할 수 ARN 있습니다. 이 예제에서는 전체 ARN 작업을 사용합니다.

```
aws ecs describe-tasks \
    --cluster MyCluster \
    --tasks arn:aws:ecs:us-east-1:123456789012:task/
MyCluster/4d590253bb114126b7afa7b58EXAMPLE
```

출력:

```
{
    "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": []
}
```

자세한 내용은 Amazon 개발자 안내서의 Amazon ECS 작업 정의를 참조하세요. ECS

예 2: 여러 태스크를 설명하는 방법

다음 describe-tasks 예시에서는 클러스터에 있는 여러 태스크의 세부 정보를 검색합니다. ID 또는 전체 태스크를 사용하여 태스크를 지정할 수 ARN 있습니다. 이 예제에서는 IDs 전체 작업을 사용합니다.

```
aws ecs describe-tasks \
    --cluster MyCluster \
    --tasks "74de0355a10a4f979ac495c14EXAMPLE" "d789e94343414c25b9f6bd59eEXAMPLE"
```

출력:

```
{
    "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": []
}
```

자세한 내용은 Amazon 개발자 안내서의 Amazon ECS 작업 정의를 참조하세요. ECS

- 자세한 API 내용은 명령 참조DescribeTasks의 섹션을 참조하세요. AWS CLI

### execute-command

다음 코드 예시에서는 execute-command을 사용하는 방법을 보여 줍니다.

AWS CLI

interactive /bin/sh 명령을 실행하려면

다음 execute-command 예제에서는 id가 인 태스크에 MyContainer 대해 라는 컨테이너에 대해 interactive /bin/sh 명령을 실행합니다arn: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"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon 개발자 안내서의 디버깅에 Amazon ECS Exec 사용을 참조하세요. ECS

- 자세한 API 내용은 명령 참조ExecuteCommand의 섹션을 참조하세요. AWS CLI

### list-account-settings

다음 코드 예시에서는 list-account-settings을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 계정의 계정 설정을 보려면

다음 list-account-settings 예제에서는 계정에 대한 유효 계정 설정을 보여줍니다.

```
aws ecs list-account-settings --effective-settings
```

출력:

```
{
    "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"
        }
    ]
}
```

예제 2: 특정 IAM 사용자 또는 IAM 역할에 대한 계정 설정을 보려면

다음 list-account-settings 예제에서는 지정된 IAM 사용자 또는 IAM 역할에 대한 계정 설정을 표시합니다.

```
aws ecs list-account-settings --principal-arn arn:aws:iam::123456789012:user/MyUser
```

출력:

```
{
    "settings": [
        {
            "name": "serviceLongArnFormat",
            "value": "enabled",
            "principalArn": "arn:aws:iam::123456789012:user/MyUser"
        }
    ]
}
```

자세한 내용은 Amazon 개발자 안내서의 Amazon 리소스 이름(ARNs) 및 IDs 섹션을 참조하세요. ECS

- 자세한 API 내용은 명령 참조ListAccountSettings의 섹션을 참조하세요. AWS CLI

### `list-attributes`

다음 코드 예시에서는 `list-attributes`을 사용하는 방법을 보여 줍니다.

AWS CLI

특정 속성을 포함하는 컨테이너 인스턴스를 나열하려면

다음 예제에서는 속성이 기본 클러스터에 있는 컨테이너 인스턴스의 `stack=production` 속성을 나열합니다.

```
aws ecs list-attributes \
    --target-type container-instance \
    --attribute-name stack \
    --attribute-value production \
    --cluster default
```

출력:

```
{
    "attributes": [
        {
            "name": "stack",
            "targetId": "arn:aws:ecs:us-west-2:130757420319:container-
instance/1c3be8ed-df30-47b4-8f1e-6e68ebd01f34",
            "value": "production"
        }
```

```
    ]
}
```

자세한 내용은 Amazon 개발자 안내서의 Amazon ECS Container Agent 구성을 참조하세요. ECS

- 자세한 API 내용은 명령 참조ListAttributes의 섹션을 참조하세요. AWS CLI

### `list-clusters`

다음 코드 예시에서는 `list-clusters`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 클러스터를 나열하는 방법

다음 `list-clusters` 예시에서는 사용 가능한 클러스터를 모두 나열합니다.

```
aws ecs list-clusters
```

출력:

```
{
    "clusterArns": [
        "arn:aws:ecs:us-west-2:123456789012:cluster/MyECSCluster1",
        "arn:aws:ecs:us-west-2:123456789012:cluster/AnotherECSCluster"
    ]
}
```

자세한 내용은 Amazon 개발자 안내서의 Amazon ECS 클러스터를 참조하세요. ECS

- 자세한 API 내용은 명령 참조ListClusters의 섹션을 참조하세요. AWS CLI

### `list-container-instances`

다음 코드 예시에서는 `list-container-instances`을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터의 컨테이너 인스턴스를 나열하려면

다음 `list-container-instances` 예제에서는 클러스터에서 사용 가능한 모든 컨테이너 인스턴스를 나열합니다.

```
aws ecs list-container-instances --cluster MyCluster
```

출력:

```
{
    "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"
    ]
}
```

자세한 내용은 Amazon 개발자 안내서의 Amazon ECS 컨테이너 인스턴스를 참조하세요. ECS

- 자세한 API 내용은 명령 참조ListContainerInstances의 섹션을 참조하세요. AWS CLI

### `list-services-by-namespace`

다음 코드 예시에서는 `list-services-by-namespace`을 사용하는 방법을 보여 줍니다.

AWS CLI

네임스페이스에 서비스를 나열하려면

다음 `list-services-by-namespace` 예제에서는 기본 리전의 지정된 네임스페이스에 대해 구성된 모든 서비스를 나열합니다.

```
aws ecs list-services-by-namespace \
    --namespace service-connect
```

출력:

```
{
    "serviceArns": [
        "arn:aws:ecs:us-west-2:123456789012:service/MyCluster/MyService",
        "arn:aws:ecs:us-west-2:123456789012:service/tutorial/service-connect-nginx-
service"
    ]
}
```

자세한 내용은 Amazon ECS 개발자 안내서의 Service Connect를 참조하세요.

- 자세한 API 내용은 명령 참조ListServicesByNamespace의 섹션을 참조하세요. AWS CLI

### `list-services`

다음 코드 예시에서는 `list-services`을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터의 서비스를 나열하는 방법

다음 `list-services` 예시에서는 클러스터에서 실행되는 서비스를 나열하는 방법을 보여줍니다.

```
aws ecs list-services --cluster MyCluster
```

출력:

```
 {
     "serviceArns": [
         "arn:aws:ecs:us-west-2:123456789012:service/MyCluster/MyService"
     ]
}
```

자세한 내용은 Amazon ECS 개발자 안내서의 서비스를 참조하세요.

- 자세한 API 내용은 명령 참조ListServices의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스의 태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 특정 클러스터의 태그를 나열합니다.

```
aws ecs list-tags-for-resource \
    --resource-arn arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster
```

출력:

```
{
    "tags": [
        {
            "key": "key1",
            "value": "value1"
        },
        {
            "key": "key2",
            "value": "value2"
        },
        {
            "key": "key3",
            "value": "value3"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### `list-task-definition-families`

다음 코드 예시에서는 `list-task-definition-families`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 등록된 태스크 정의 패밀리를 나열하려면

다음 `list-task-definition-families` 예제에서는 등록된 태스크 정의 패밀리를 모두 나열합니다.

```
aws ecs list-task-definition-families
```

출력:

```
{
    "families": [
        "node-js-app",
        "web-timer",
        "hpcc",
        "hpcc-c4-8xlarge"
    ]
```

```
}
```

예제 2: 등록된 태스크 정의 패밀리를 필터링하려면

다음 `list-task-definition-families` 예제에서는 “hpcc”로 시작하는 태스크 정의 개정을 나열합니다.

```
aws ecs list-task-definition-families --family-prefix hpcc
```

출력:

```
{
    "families": [
        "hpcc",
        "hpcc-c4-8xlarge"
    ]
}
```

자세한 내용은 Amazon ECS 개발자 안내서의 작업 정의 파라미터를 참조하세요.

- 자세한 API 내용은 명령 참조ListTaskDefinitionFamilies의 섹션을 참조하세요. AWS CLI

### `list-task-definitions`

다음 코드 예시에서는 `list-task-definitions`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 등록된 작업 정의를 나열하려면

다음 `list-task-definitions` 예제에서는 등록된 모든 태스크 정의를 나열합니다.

```
aws ecs list-task-definitions
```

출력:

```
{
    "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"
    ]
}
```

예제 2: 패밀리에 등록된 태스크 정의를 나열하려면

다음 list-task-definitions 예제에서는 지정된 패밀리의 태스크 정의 개정을 나열합니다.

```
aws ecs list-task-definitions --family-prefix wordpress
```

출력:

```
{
    "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"
    ]
}
```

자세한 내용은 Amazon 개발자 안내서의 Amazon ECS 작업 정의를 참조하세요. ECS

- 자세한 API 내용은 명령 참조ListTaskDefinitions의 섹션을 참조하세요. AWS CLI

### list-tasks

다음 코드 예시에서는 `list-tasks`을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: 클러스터의 태스크를 나열하는 방법

다음 `list-tasks` 예시에서는 클러스터의 모든 태스크를 나열합니다.

```
aws ecs list-tasks --cluster default
```

출력:

```
{
    "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"
    ]
}
```

예 2: 특정 컨테이너 인스턴스의 태스크를 나열하는 방법

다음 `list-tasks` 예제에서는 컨테이너 인스턴스를 필터UUID로 사용하여 컨테이너 인스턴스의 작업을 나열합니다.

```
aws ecs list-tasks --cluster default --container-instance a1b2c3d4-5678-90ab-
cdef-33333EXAMPLE
```

출력:

```
{
    "taskArns": [
        "arn:aws:ecs:us-west-2:123456789012:task/a1b2c3d4-5678-90ab-
cdef-44444EXAMPLE"
    ]
}
```

자세한 내용은 Amazon 개발자 안내서의 Amazon ECS 작업 정의를 참조하세요. ECS

- 자세한 API 내용은 명령 참조ListTasks의 섹션을 참조하세요. AWS CLI

### put-account-setting-default

다음 코드 예시에서는 `put-account-setting-default`을 사용하는 방법을 보여 줍니다.

AWS CLI

기본 계정 설정을 수정하려면

다음 `put-account-setting-default` 예제에서는 계정의 모든 IAM 사용자 또는 역할에 대한 기본 계정 설정을 수정합니다. IAM 사용자 또는 역할이 이러한 설정을 명시적으로 재정의하지 않는 한 이러한 변경 사항은 전체 AWS 계정에 적용됩니다.

```
aws ecs put-account-setting-default --name serviceLongArnFormat --value enabled
```

출력:

```
{
    "setting": {
        "name": "serviceLongArnFormat",
        "value": "enabled",
        "principalArn": "arn:aws:iam::123456789012:root"
    }
}
```

자세한 내용은 Amazon 개발자 안내서의 Amazon 리소스 이름(ARNs) 및 IDs 섹션을 참조하세요. ECS

- 자세한 API 내용은 명령 참조PutAccountSettingDefault의 섹션을 참조하세요. AWS CLI

### put-account-setting

다음 코드 예시에서는 put-account-setting을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 사용자 계정의 계정 설정을 수정하려면

다음 put-account-setting 예제에서는 IAM 사용자 serviceLongArnFormat 계정에 대한 계정 설정을 활성화합니다.

```
aws ecs put-account-setting --name serviceLongArnFormat --value enabled
```

출력:

```
{
    "setting": {
        "name": "serviceLongArnFormat",
        "value": "enabled",
        "principalArn": "arn:aws:iam::130757420319:user/your_username"
    }
}
```

자세한 내용은 Amazon ECS 개발자 안내서의 계정 설정 수정을 참조하세요.

- 자세한 API 내용은 명령 참조PutAccountSetting의 섹션을 참조하세요. AWS CLI

### put-account-settings

다음 코드 예시에서는 put-account-settings을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 사용자 또는 IAM 역할에 대한 계정 설정을 수정하려면

다음 put-account-setting 예제에서는 지정된 IAM 사용자 또는 IAM 역할에 대한 계정 설정을 수정합니다.

```
aws ecs put-account-setting \
    --name serviceLongArnFormat \
    --value enabled \
    --principal-arn arn:aws:iam::123456789012:user/MyUser
```

출력:

```
{
    "setting": {
        "name": "serviceLongArnFormat",
        "value": "enabled",
        "principalArn": "arn:aws:iam::123456789012:user/MyUser"
    }
}
```

- 자세한 API 내용은 명령 참조PutAccountSettings의 섹션을 참조하세요. AWS CLI

### put-attributes

다음 코드 예시에서는 put-attributes을 사용하는 방법을 보여 줍니다.

AWS CLI

속성을 생성하고 Amazon ECS 리소스와 연결하려면

다음은 이름 스택과 값 프로덕션이 있는 속성을 컨테이너 인스턴스에 put-attributes 적용합니다.

```
aws ecs put-attributes \
```

```
    --attributes name=stack,value=production,targetId=arn:aws:ecs:us-
west-2:130757420319:container-instance/1c3be8ed-df30-47b4-8f1e-6e68ebd01f34
```

출력:

```
{
    "attributes": [
        {
            "name": "stack",
            "targetId": "arn:aws:ecs:us-west-2:130757420319:container-
instance/1c3be8ed-df30-47b4-8f1e-6e68ebd01f34",
            "value": "production"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조PutAttributes의 섹션을 참조하세요. AWS CLI

### put-cluster-capacity-providers

다음 코드 예시에서는 put-cluster-capacity-providers을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 클러스터에 기존 용량 공급자를 추가하려면

다음 put-cluster-capacity-providers 예제에서는 클러스터에 기존 용량 공급자를 추가합니다. create-capacity-provider 명령을 사용하여 용량 공급자를 생성합니다. 이 describe-clusters 명령은 현재 용량 공급자와 클러스터와 연결된 기본 용량 공급자 전략을 설명하는 데 사용됩니다. 클러스터에 새 용량 공급자를 추가할 때는 클러스터와 연결하려는 새 용량 공급자 외에 기존 용량 공급자를 모두 지정해야 합니다. 클러스터와 연결할 기본 용량 공급자 전략도 지정해야 합니다. 이 예제에서는 MyCluster 클러스터에 연결된 MyCapacityProvider1 용량 공급자가 있으며, 두 용량 공급자 간에 작업이 균등하게 분산되도록 MyCapacityProvider2 용량 공급자를 추가하고 기본 용량 공급자 전략에 포함시키려는 경우

```
aws ecs put-cluster-capacity-providers \
    --cluster MyCluster \
    --capacity-providers MyCapacityProvider1 MyCapacityProvider2 \
    --default-capacity-provider-
strategy capacityProvider=MyCapacityProvider1,weight=1 capacityProvider=MyCapacityProvider2,
```

출력:

```
{
    "cluster": {
        "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster",
        "clusterName": "MyCluster",
        "status": "ACTIVE",
        "registeredContainerInstancesCount": 0,
        "runningTasksCount": 0,
        "pendingTasksCount": 0,
        "activeServicesCount": 0,
        "statistics": [],
        "tags": [],
        "settings": [
            {
                "name": "containerInsights",
                "value": "enabled"
            }
        ],
        "capacityProviders": [
            "MyCapacityProvider1",
            "MyCapacityProvider2"
        ],
        "defaultCapacityProviderStrategy": [
            {
                "capacityProvider": "MyCapacityProvider1",
                "weight": 1,
                "base": 0
            },
            {
                "capacityProvider": "MyCapacityProvider2",
                "weight": 1,
                "base": 0
            }
        ],
        "attachments": [
           {
                "id": "0fb0c8f4-6edd-4de1-9b09-17e470ee1918",
                "type": "as_policy",
                "status": "ACTIVE",
                "details": [
                    {
                        "name": "capacityProviderName",
                        "value": "MyCapacityProvider1"
```

```
                    },
                    {
                        "name": "scalingPolicyName",
                        "value": "ECSManagedAutoScalingPolicy-a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111"
                    }
                ]
            },
            {
                "id": "ae592060-2382-4663-9476-b015c685593c",
                "type": "as_policy",
                "status": "ACTIVE",
                "details": [
                    {
                        "name": "capacityProviderName",
                        "value": "MyCapacityProvider2"
                    },
                    {
                        "name": "scalingPolicyName",
                        "value": "ECSManagedAutoScalingPolicy-a1b2c3d4-5678-90ab-
cdef-EXAMPLE22222"
                    }
                ]
            }
        ],
        "attachmentsStatus": "UPDATE_IN_PROGRESS"
    }
}
```

자세한 내용은 Amazon ECS 개발자 안내서의 클러스터 용량 공급자를 참조하세요.

예제 2: 클러스터에서 용량 공급자를 제거하려면

다음 put-cluster-capacity-providers 예제에서는 클러스터에서 용량 공급자를 제거합니다. 이 describe-clusters 명령은 클러스터와 연결된 현재 용량 공급자를 설명하는 데 사용됩니다. 클러스터에서 용량 공급자를 제거할 때는 클러스터와 연결하려는 용량 공급자와 클러스터와 연결할 기본 용량 공급자 전략을 지정해야 합니다. 이 예제에서는 클러스터에 MyCapacityProvider1 및 MyCapacityProvider2 용량 공급자가 연결되어 있고 MyCapacityProvider2 용량 공급자를 제거하려고 하므로 업데이트된 기본 용량 공급자 전략과 함께 명령MyCapacityProvider1에만 를 지정합니다.

```
aws ecs put-cluster-capacity-providers \
```

```
    --cluster MyCluster \
    --capacity-providers MyCapacityProvider1 \
    --default-capacity-provider-
strategy capacityProvider=MyCapacityProvider1,weight=1,base=0
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon ECS 개발자 안내서의 클러스터 용량 공급자를 참조하세요.

예제 3: 클러스터에서 모든 용량 공급자 제거

다음 `put-cluster-capacity-providers` 예제에서는 클러스터에서 기존 용량 공급자를 모두 제거합니다.

```
aws ecs put-cluster-capacity-providers \
    --cluster MyCluster \
    --capacity-providers [] \
    --default-capacity-provider-strategy []
```

출력:

```
{
    "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-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"
    }
}
```

자세한 내용은 Amazon ECS 개발자 안내서의 클러스터 용량 공급자를 참조하세요.

- 자세한 API 내용은 명령 참조PutClusterCapacityProviders의 섹션을 참조하세요. AWS CLI

### register-task-definition

다음 코드 예시에서는 register-task-definition을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: JSON 파일에 태스크 정의를 등록하려면

다음 register-task-definition 예제에서는 지정된 패밀리에 태스크 정의를 등록합니다. 컨테이너 정의는 지정된 파일 위치에 JSON 형식으로 저장됩니다.

```
aws ecs register-task-definition \
    --cli-input-json file://<path_to_json_file>/sleep360.json
```

sleep360.json의 콘텐츠:

```
{
    "containerDefinitions": [
        {
            "name": "sleep",
            "image": "busybox",
            "cpu": 10,
            "command": [
                "sleep",
                "360"
            ],
            "memory": 10,
            "essential": true
```

```
        }
    ],
    "family": "sleep360"
}
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 Amazon ECS 개발자 안내서의 작업 정의 예제를 참조하세요.

예제 2: JSON 문자열 파라미터로 작업 정의를 등록하려면

다음 register-task-definition 예제에서는 이스케이프된 큰따옴표가 있는 JSON 문자열 파라미터로 제공된 컨테이너 정의를 사용하여 작업 정의를 등록합니다.

```
aws ecs register-task-definition \
    --family sleep360 \
    --container-definitions "[{\"name\":\"sleep\",\"image\":\"busybox\",\"cpu\":10,
\"command\":[\"sleep\",\"360\"],\"memory\":10,\"essential\":true}]"
```

출력은 이전 예제와 동일합니다.

자세한 내용은 Amazon ECS 개발자 안내서의 작업 정의 생성을 참조하세요.

- 자세한 API 내용은 명령 참조RegisterTaskDefinition의 섹션을 참조하세요. AWS CLI

### run-task

다음 코드 예시에서는 run-task을 사용하는 방법을 보여 줍니다.

AWS CLI

기본 클러스터에서 작업을 실행하려면

다음 run-task 예제에서는 기본 클러스터에서 작업을 실행하고 클라이언트 토큰을 사용합니다.

```
aws ecs run-task \
    --cluster default \
    --task-definition sleep360:1 \
    --client-token 550e8400-e29b-41d4-a716-446655440000
```

출력:

```
{
    "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": []
}
```

자세한 내용은 Amazon ECS 개발자 안내서의 작업 실행을 참조하세요.

- 자세한 API 내용은 명령 참조RunTask의 섹션을 참조하세요. AWS CLI

### start-task

다음 코드 예시에서는 start-task을 사용하는 방법을 보여 줍니다.

AWS CLI

새 작업을 시작하려면

다음은 기본 클러스터의 지정된 컨테이너 인스턴스에서 작업 정의의 최신 개정을 사용하여 sleep360 작업을 start-task 시작합니다.

```
aws ecs start-task \
    --task-definition sleep360 \
    --container-instances 765936fadbdd46b5991a4bd70c2a43d4
```

출력:

```
{
    "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": []
}
```

- 자세한 API 내용은 명령 참조StartTask의 섹션을 참조하세요. AWS CLI

### stop-task

다음 코드 예시에서는 stop-task을 사용하는 방법을 보여 줍니다.

AWS CLI

작업을 중지하려면

다음은 지정된 태스크가 기본 클러스터에서 실행되지 않도록 stop-task 중지합니다.

```
aws ecs stop-task \
    --task 666fdccc2e2d4b6894dd422f4eeee8f8
```

출력:

```
{
```

```
    "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": []
    }
}
```

- 자세한 API 내용은 명령 참조StopTask의 섹션을 참조하세요. AWS CLI

### `tag-resource`

다음 코드 예시에서는 `tag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 태그를 지정하려면

다음 tag-resource 예제에서는 지정된 리소스에 단일 태그를 추가합니다.

```
aws ecs tag-resource \
    --resource-arn arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster
    --tags key=key1,value=value1
```

이 명령은 출력을 생성하지 않습니다.

리소스에 여러 태그를 추가하려면

다음 tag-resource 예제에서는 지정된 리소스에 여러 태그를 추가합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### untag-resource

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에서 태그를 제거하려면

다음 untag-resource 예제에서는 지정된 리소스에서 나열된 태그를 제거합니다.

```
aws ecs untag-resource \
    --resource-arn arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster \
    --tag-keys key1,key2
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-cluster-settings

다음 코드 예시에서는 update-cluster-settings을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터의 설정을 수정하려면

다음 update-cluster-settings 예제에서는 default 클러스터에 대한 CloudWatch Container Insights를 활성화합니다.

```
aws ecs update-cluster-settings \
    --cluster default \
    --settings name=containerInsights,value=enabled
```

출력:

```
{
    "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"
            }
        ]
    }
}
```

자세한 내용은 Amazon ECS 개발자 안내서의 계정 설정 수정을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateClusterSettings의 섹션을 참조하세요. AWS CLI

### update-container-agent

다음 코드 예시에서는 update-container-agent을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon 컨테이너 인스턴스에서 ECS 컨테이너 에이전트를 업데이트하려면

다음 update-container-agent 예제에서는 기본 클러스터의 지정된 컨테이너 인스턴스에서 컨테이너 에이전트를 업데이트합니다.

```
aws ecs update-container-agent --cluster default --container-
instance a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

출력:

```
{
    "containerInstance": {
        "status": "ACTIVE",
...
        "agentUpdateStatus": "PENDING",
        "versionInfo": {
            "agentVersion": "1.0.0",
            "agentHash": "4023248",
            "dockerVersion": "DockerVersion: 1.5.0"
        }
    }
}
```

자세한 내용은 Amazon 개발자 안내서의 Amazon ECS 컨테이너 에이전트 업데이트를 참조하세요. ECS

- 자세한 API 내용은 명령 참조UpdateContainerAgent의 섹션을 참조하세요. AWS CLI

### update-container-instances-state

다음 코드 예시에서는 update-container-instances-state을 사용하는 방법을 보여 줍니다.

AWS CLI

컨테이너 인스턴스의 상태를 업데이트하려면

다음은 클러스터에서 DRAINING 제거할 지정된 컨테이너 인스턴스의 상태를 등록된 클러스터에서 update-container-instances-state 업데이트합니다.

```
aws ecs update-container-instances-state \
```

```
    --container-instances 765936fadbdd46b5991a4bd70c2a43d4 \
    --status DRAINING
```

출력:

```
{
    "containerInstances": [
        {
            "containerInstanceArn": "arn:aws:ecs:us-west-2:130757420319:container-
instance/default/765936fadbdd46b5991a4bd70c2a43d4",
            "ec2InstanceId": "i-013d87ffbb4d513bf",
            "version": 4390,
            "versionInfo": {
                "agentVersion": "1.29.0",
                "agentHash": "a190a73f",
                "dockerVersion": "DockerVersion:18.06.1-ce"
            },
            "remainingResources": [
                {
                    "name": "CPU",
                    "type": "INTEGER",
                    "doubleValue": 0,
                    "longValue": 0,
                    "integerValue": 1536
                },
                {
                    "name": "MEMORY",
                    "type": "INTEGER",
                    "doubleValue": 0,
                    "longValue": 0,
                    "integerValue": 2681
                },
                {
                    "name": "PORTS",
                    "type": "STRINGSET",
                    "doubleValue": 0,
                    "longValue": 0,
                    "integerValue": 0,
                    "stringSetValue": [
                        "22",
                        "2376",
                        "2375",
                        "51678",
```

```
                    "51679"
                ]
            },
            {
                "name": "PORTS_UDP",
                "type": "STRINGSET",
                "doubleValue": 0,
                "longValue": 0,
                "integerValue": 0,
                "stringSetValue": []
            }
        ],
        "registeredResources": [
            {
                "name": "CPU",
                "type": "INTEGER",
                "doubleValue": 0,
                "longValue": 0,
                "integerValue": 2048
            },
            {
                "name": "MEMORY",
                "type": "INTEGER",
                "doubleValue": 0,
                "longValue": 0,
                "integerValue": 3705
            },
            {
                "name": "PORTS",
                "type": "STRINGSET",
                "doubleValue": 0,
                "longValue": 0,
                "integerValue": 0,
                "stringSetValue": [
                    "22",
                    "2376",
                    "2375",
                    "51678",
                    "51679"
                ]
            },
            {
                "name": "PORTS_UDP",
                "type": "STRINGSET",
```

```
                "doubleValue": 0,
                "longValue": 0,
                "integerValue": 0,
                "stringSetValue": []
            }
        ],
        "status": "DRAINING",
        "agentConnected": true,
        "runningTasksCount": 2,
        "pendingTasksCount": 0,
        "attributes": [
            {
                "name": "ecs.capability.secrets.asm.environment-variables"
            },
            {
                "name": "ecs.capability.branch-cni-plugin-version",
                "value": "e0703516-"
            },
            {
                "name": "ecs.ami-id",
                "value": "ami-00e0090ac21971297"
            },
            {
                "name": "ecs.capability.secrets.asm.bootstrap.log-driver"
            },
            {
                "name": "com.amazonaws.ecs.capability.logging-driver.none"
            },
            {
                "name": "ecs.capability.ecr-endpoint"
            },
            {
                "name": "ecs.capability.docker-plugin.local"
            },
            {
                "name": "ecs.capability.task-cpu-mem-limit"
            },
            {
                "name": "ecs.capability.secrets.ssm.bootstrap.log-driver"
            },
            {
                "name": "com.amazonaws.ecs.capability.docker-remote-api.1.30"
            },
            {
```

```
            "name": "com.amazonaws.ecs.capability.docker-remote-api.1.31"
        },
        {
            "name": "com.amazonaws.ecs.capability.docker-remote-api.1.32"
        },
        {
            "name": "ecs.availability-zone",
            "value": "us-west-2c"
        },
        {
            "name": "ecs.capability.aws-appmesh"
        },
        {
            "name": "com.amazonaws.ecs.capability.logging-driver.awslogs"
        },
        {
            "name": "com.amazonaws.ecs.capability.docker-remote-api.1.24"
        },
        {
            "name": "ecs.capability.task-eni-trunking"
        },
        {
            "name": "com.amazonaws.ecs.capability.docker-remote-api.1.25"
        },
        {
            "name": "com.amazonaws.ecs.capability.docker-remote-api.1.26"
        },
        {
            "name": "com.amazonaws.ecs.capability.docker-remote-api.1.27"
        },
        {
            "name": "com.amazonaws.ecs.capability.docker-remote-api.1.28"
        },
        {
            "name": "com.amazonaws.ecs.capability.privileged-container"
        },
        {
            "name": "com.amazonaws.ecs.capability.docker-remote-api.1.29"
        },
        {
            "name": "ecs.cpu-architecture",
            "value": "x86_64"
        },
        {
```

```
                    "name": "com.amazonaws.ecs.capability.ecr-auth"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.20"
                },
                {
                    "name": "ecs.os-type",
                    "value": "linux"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.21"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.22"
                },
                {
                    "name": "ecs.capability.task-eia"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.23"
                },
                {
                    "name": "ecs.capability.private-registry-
authentication.secretsmanager"
                },
                {
                    "name": "com.amazonaws.ecs.capability.logging-driver.syslog"
                },
                {
                    "name": "com.amazonaws.ecs.capability.logging-driver.json-file"
                },
                {
                    "name": "ecs.capability.execution-role-awslogs"
                },
                {
                    "name": "ecs.vpc-id",
                    "value": "vpc-1234"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.17"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.18"
                },
```

```
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.19"
                },
                {
                    "name": "ecs.capability.task-eni"
                },
                {
                    "name": "ecs.capability.execution-role-ecr-pull"
                },
                {
                    "name": "ecs.capability.container-health-check"
                },
                {
                    "name": "ecs.subnet-id",
                    "value": "subnet-1234"
                },
                {
                    "name": "ecs.instance-type",
                    "value": "c5.large"
                },
                {
                    "name": "com.amazonaws.ecs.capability.task-iam-role-network-
host"
                },
                {
                    "name": "ecs.capability.container-ordering"
                },
                {
                    "name": "ecs.capability.cni-plugin-version",
                    "value": "91ccefc8-2019.06.0"
                },
                {
                    "name": "ecs.capability.pid-ipc-namespace-sharing"
                },
                {
                    "name": "ecs.capability.secrets.ssm.environment-variables"
                },
                {
                    "name": "com.amazonaws.ecs.capability.task-iam-role"
                }
            ],
            "registeredAt": 1560788724.507,
            "attachments": [],
            "tags": []
```

```
        }
    ],
    "failures": []
}
```

- 자세한 API 내용은 명령 참조UpdateContainerInstancesState의 섹션을 참조하세요. AWS CLI

### update-service-primary-task-set

다음 코드 예시에서는 update-service-primary-task-set을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스의 기본 태스크 세트를 업데이트하려면

다음 update-service-primary-task-set 예제에서는 지정된 서비스에 대한 기본 태스크 세트를 업데이트합니다.

```
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
```

출력:

```
{
    "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
    }
}
```

- 자세한 API 내용은 명령 참조UpdateServicePrimaryTaskSet의 섹션을 참조하세요. AWS CLI

### update-service

다음 코드 예시에서는 update-service을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: 서비스에 사용되는 태스크 정의를 변경하는 방법

다음 update-service 예시에서는 amazon-ecs-sample 태스크 정의를 사용하도록 my-http-service 서비스를 업데이트합니다.

```
aws ecs update-service --service my-http-service --task-definition amazon-ecs-sample
```

예 2: 서비스의 태스크 수를 변경하는 방법

다음 update-service 예시에서는 my-http-service 서비스의 원하는 태스크 수를 3으로 업데이트합니다.

```
aws ecs update-service --service my-http-service --desired-count 3
```

자세한 내용은 Amazon ECS 개발자 안내서의 서비스 업데이트를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateService의 섹션을 참조하세요. AWS CLI

### update-task-set

다음 코드 예시에서는 update-task-set을 사용하는 방법을 보여 줍니다.

AWS CLI

작업 세트를 업데이트하려면

다음 update-task-set 예제에서는 작업 세트를 업데이트하여 스케일을 조정합니다.

```
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
```

출력:

```
{
    "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
    }
}
```

- 자세한 API 내용은 명령 참조UpdateTaskSet의 섹션을 참조하세요. AWS CLI

## 를 사용한 Amazon EFS 예제 AWS CLI

다음 코드 예제에서는 Amazon 에서 를 사용하여 작업을 수행하고 일반적인 시나리오 AWS Command Line Interface 를 구현하는 방법을 보여줍니다EFS.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `create-file-system`

다음 코드 예시에서는 `create-file-system`을 사용하는 방법을 보여 줍니다.

AWS CLI

암호화된 파일 시스템을 생성하려면

다음 `create-file-system` 예제에서는 기본 를 사용하여 암호화된 파일 시스템을 생성합니다 CMK. 태그도 추가합니다`Name=my-file-system`.

```
aws efs create-file-system \
    --performance-mode generalPurpose \
    --throughput-mode bursting \
    --encrypted \
    --tags Key=Name,Value=my-file-system
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Elastic EFS File System 사용 설명서의 Amazon 파일 시스템 생성을 참조하세요. Amazon Elastic File System

- 자세한 API 내용은 명령 참조CreateFileSystem의 섹션을 참조하세요. AWS CLI

### `create-mount-target`

다음 코드 예시에서는 `create-mount-target`을 사용하는 방법을 보여 줍니다.

AWS CLI

탑재 대상을 생성하려면

다음 `create-mount-target` 예제에서는 지정된 파일 시스템에 대한 탑재 대상을 생성합니다.

```
aws efs create-mount-target \
    --file-system-id fs-c7a0456e \
    --subnet-id subnet-02bf4c428bexample \
    --security-groups sg-068f739363example
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon Elastic File System 사용 설명서의 탑재 대상 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateMountTarget의 섹션을 참조하세요. AWS CLI

### `delete-file-system`

다음 코드 예시에서는 `delete-file-system`을 사용하는 방법을 보여 줍니다.

AWS CLI

파일 시스템을 삭제하려면

다음 delete-file-system 예제에서는 지정된 파일 시스템을 삭제합니다.

```
aws efs delete-file-system \
    --file-system-id fs-c7a0456e
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Elastic EFS File System 사용 설명서의 Amazon 파일 시스템 삭제를 참조하세요. Amazon Elastic File System

- 자세한 API 내용은 명령 참조DeleteFileSystem의 섹션을 참조하세요. AWS CLI

### delete-mount-target

다음 코드 예시에서는 delete-mount-target을 사용하는 방법을 보여 줍니다.

AWS CLI

탑재 대상을 삭제하려면

다음 delete-mount-target 예제에서는 지정된 탑재 대상을 삭제합니다.

```
aws efs delete-mount-target \
    --mount-target-id fsmt-f9a14450
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Elastic File System 사용 설명서의 탑재 대상 생성을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteMountTarget의 섹션을 참조하세요. AWS CLI

### describe-file-systems

다음 코드 예시에서는 describe-file-systems을 사용하는 방법을 보여 줍니다.

AWS CLI

파일 시스템을 설명하려면

다음 describe-file-systems 예제에서는 지정된 파일 시스템을 설명합니다.

```
aws efs describe-file-systems \
    --file-system-id fs-c7a0456e
```

출력:

```
{
    "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"
                }
            ]
        }
    ]
}
```

자세한 내용은 Amazon Elastic EFS File System 사용 설명서의 Amazon 파일 시스템 관리를 참조하세요. Amazon Elastic File System

- 자세한 API 내용은 명령 참조DescribeFileSystems의 섹션을 참조하세요. AWS CLI

### describe-mount-targets

다음 코드 예시에서는 describe-mount-targets을 사용하는 방법을 보여 줍니다.

AWS CLI

탑재 대상을 설명하려면

다음 describe-mount-targets 예제에서는 지정된 탑재 대상을 설명합니다.

```
aws efs describe-mount-targets \
    --mount-target-id fsmt-f9a14450
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Elastic File System 사용 설명서의 탑재 대상 생성을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeMountTargets의 섹션을 참조하세요. AWS CLI

### describe-tags

다음 코드 예시에서는 describe-tags을 사용하는 방법을 보여 줍니다.

AWS CLI

파일 시스템의 태그를 설명하려면

다음 `describe-tags` 예제에서는 지정된 파일 시스템의 태그를 설명합니다.

```
aws efs describe-tags \
    --file-system-id fs-c7a0456e
```

출력:

```
{
    "Tags": [
        {
            "Key": "Name",
            "Value": "my-file-system"
        },
        {
            "Key": "Department",
            "Value": "Business Intelligence"
        }
    ]
}
```

자세한 내용은 Amazon Elastic File System File System 사용 설명서의 파일 시스템 태그 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeTags의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스의 태그를 검색하려면

다음 `list-tags-for-resource` 예제에서는 지정된 파일 시스템과 연결된 태그를 검색합니다.

```
aws efs list-tags-for-resource \
    --resource-id fs-c7a0456e
```

출력:

```
{
    "Tags": [
        {
            "Key": "Name",
            "Value": "my-file-system"
        },
        {
            "Key": "Department",
            "Value": "Business Intelligence"
        }
    ]
}
```

자세한 내용은 Amazon Elastic File System File System 사용 설명서의 파일 시스템 태그 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### `tag-resource`

다음 코드 예시에서는 `tag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 태그를 지정하려면

다음 `tag-resource` 예제에서는 `Department=Business Intelligence` 지정된 파일 시스템에 태그를 추가합니다.

```
aws efs tag-resource \
    --resource-id fs-c7a0456e \
    --tags Key=Department,Value="Business Intelligence"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Elastic File System File System 사용 설명서의 파일 시스템 태그 관리를 참조하세요.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

**`untag-resource`**

다음 코드 예시에서는 `untag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에서 태그를 제거하려면

다음 `untag-resource` 예제에서는 지정된 파일 시스템에서 태그 키가 있는 `Department` 태그를 제거합니다.

```
aws efs untag-resource \
    --resource-id fs-c7a0456e \
    --tag-keys Department
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Elastic File System File System 사용 설명서의 파일 시스템 태그 관리를 참조하세요.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

## 를 사용한 Amazon EKS 예제 AWS CLI

다음 코드 예제에서는 Amazon 와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다EKS.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### `associate-encryption-config`

다음 코드 예시에서는 associate-encryption-config을 사용하는 방법을 보여 줍니다.

AWS CLI

암호화 구성을 기존 클러스터에 연결하려면

다음 associate-encryption-config 예제에서는 암호화가 아직 활성화되지 않은 기존 EKS 클러스터에서 의 암호화를 활성화합니다.

```
aws eks associate-encryption-config \
    --cluster-name my-eks-cluster \
    --encryption-config '[{"resources":["secrets"],"provider":
{"keyArn":"arn:aws:kms:region-code:account:key/key"}}]'
```

출력:

```
{
    "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": []
    }
}
```

자세한 내용은 Amazon EKS 사용 설명서의 기존 클러스터에서 보안 암호 암호화 활성화를 참조하세요.

- 자세한 API 내용은 명령 참조AssociateEncryptionConfig의 섹션을 참조하세요. AWS CLI

### associate-identity-provider-config

다음 코드 예시에서는 associate-identity-provider-config을 사용하는 방법을 보여 줍니다.

AWS CLI

자격 증명 공급자를 Amazon EKS 클러스터에 연결

다음 associate-identity-provider-config 예제에서는 자격 증명 공급자를 Amazon EKS 클러스터에 연결합니다.

```
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
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon EKS 사용 설명서의 OpenID Connect 자격 증명 공급자에서 클러스터 사용자 인증 - OIDC 자격 증명 공급자 연결을 참조하세요.

- 자세한 API 내용은 명령 참조AssociateIdentityProviderConfig의 섹션을 참조하세요. AWS CLI

### `create-addon`

다음 코드 예시에서는 `create-addon`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 각 EKS 클러스터 버전에 대한 기본 호환성 버전으로 Amazon EKS 추가 기능을 생성하려면

다음 `create-addon` 예제 명령은 각 EKS 클러스터 버전에 대한 기본 호환성 버전을 사용하여 Amazon EKS 추가 기능을 생성합니다.

```
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
```

출력:

```
{
    "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": {}
    }
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon EKS 추가 기능 관리 - 추가 기능 생성을 참조하세요. EKS

예제 2: 특정 EKS 추가 기능 버전으로 Amazon 추가 기능을 생성하려면

다음 create-addon 예제 명령은 특정 EKS 추가 기능 버전을 사용하여 Amazon 추가 기능을 생성합니다.

```
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
```

출력:

```
{
    "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": {}
    }
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon EKS 추가 기능 관리 - 추가 기능 생성을 참조하세요. EKS

예제 3: 사용자 지정 구성 값으로 Amazon EKS 추가 기능을 생성하고 충돌 세부 정보를 해결하려면

다음 create-addon 예제 명령은 사용자 지정 구성 값을 사용하여 Amazon EKS 추가 기능을 생성하고 충돌 세부 정보를 해결합니다.

```
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
```

출력:

```
{
    "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\"}}}"
    }
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon EKS 추가 기능 관리 - 추가 기능 생성을 참조하세요. EKS

예제 4: 사용자 지정 JSON 구성 값 파일로 Amazon EKS 추가 기능 생성

다음 `create-addon` 예제 명령은 사용자 지정 구성 값을 사용하여 Amazon EKS 추가 기능을 생성하고 충돌 세부 정보를 해결합니다.

```
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"}'
```

configuration-values.json의 콘텐츠:

```
{
    "resources": {
        "limits": {
            "cpu": "150m"
        }
    },
    "env": {
        "AWS_VPC_K8S_CNI_LOGLEVEL": "ERROR"
    }
}
```

출력:

```
{
    "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-fb32-6a7be19ececd",
        "createdAt": "2024-03-14T13:10:51.763000-04:00",
        "modifiedAt": "2024-03-14T13:10:51.777000-04:00",
        "serviceAccountRoleArn": "arn:aws:iam::111122223333:role/role-name",
        "tags": {
            "eks-addon-key-1": "value-1",
            "eks-addon-key-2": "value-2"
        },
        "configurationValues": "{\n    \"resources\": {\n        \"limits\":
 {\n            \"cpu\": \"150m\"\n        }\n    },\n    \"env\": {\n
 \"AWS_VPC_K8S_CNI_LOGLEVEL\": \"ERROR\"\n    }\n}"
    }
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon EKS 추가 기능 관리 - 추가 기능 생성을 참조하세요. EKS

예제 5: 사용자 지정 YAML 구성 값 파일로 Amazon EKS 추가 기능 생성

다음 create-addon 예제 명령은 사용자 지정 구성 값을 사용하여 Amazon EKS 추가 기능을 생성하고 충돌 세부 정보를 해결합니다.

```
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"}'
```

configuration-values.yaml의 콘텐츠:

```
resources:
    limits:
        cpu: '100m'
env:
    AWS_VPC_K8S_CNI_LOGLEVEL: 'DEBUG'
```

출력:

```
{
    "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'"
```

```
    }
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon EKS 추가 기능 관리 - 추가 기능 생성을 참조하세요. EKS

- 자세한 API 내용은 명령 참조CreateAddon의 섹션을 참조하세요. AWS CLI

### `create-cluster`

다음 코드 예시에서는 `create-cluster`을 사용하는 방법을 보여 줍니다.

AWS CLI

새로운 클러스터를 생성하는 방법

이 예시 명령은 기본 리전에 이름이 `prod`인 클러스터를 생성합니다.

명령:

```
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
```

출력:

```
{
    "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": {}
    }
}
```

프라이빗 엔드포인트 액세스 및 로깅이 활성화된 새 클러스터를 생성하는 방법

이 예시 명령은 퍼블릭 엔드포인트 액세스가 비활성화되고, 프라이빗 엔드포인트 액세스가 활성화되고, 모든 로깅 유형이 활성화된 상태로 기본 리전에 이름이 example인 클러스터를 생성합니다.

명령:

```
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}]}'
```

출력:

```
{
    "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"
    }
}
```

- 자세한 API 내용은 명령 참조CreateCluster의 섹션을 참조하세요. AWS CLI

### `create-fargate-profile`

다음 코드 예시에서는 `create-fargate-profile`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 네임스페이스가 있는 선택기에 대한 EKS Fargate 프로파일 생성

다음 `create-fargate-profile` 예제에서는 네임스페이스가 있는 선택기에 대한 EKS Fargate 프로파일을 생성합니다.

```
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"}]'
```

출력:

```
{
    "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": {}
    }
}
```

자세한 내용은 Amazon EKS 사용 설명서의 AWS Fargate 프로필 - Fargate 프로필 생성을 참조하세요.

예제 2: 네임스페이스와 레이블이 있는 선택기에 대한 EKS Fargate 프로파일 생성

다음 `create-fargate-profile` 예제에서는 네임스페이스와 레이블이 있는 선택기에 대한 EKS Fargate 프로파일을 생성합니다.

```
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"}}]'
```

출력:

```
{
    "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": {}
    }
}
```

자세한 내용은 Amazon EKS 사용 설명서의 AWS Fargate 프로필 - Fargate 프로필 생성을 참조하세요.

예제 3: 포드를 시작할 서브넷과 함께 네임스페이스와 레이블이 있는 선택기IDs에 대한 EKS Fargate 프로파일 생성.

다음 `create-fargate-profile` 예제에서는 포드를 시작할 IDs 서브넷과 함께 네임스페이스와 레이블이 있는 선택기에 대한 EKS Fargate 프로파일을 생성합니다.

```
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"]'
```

출력:

```
{
    "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": {}
    }
}
```

자세한 내용은 Amazon EKS 사용 설명서의 AWS Fargate 프로필 - Fargate 프로필 생성을 참조하세요.

예제 4: 포드를 시작할 서브넷과 함께 여러 네임스페이스 및 레이블이 있는 선택기IDs의 EKS Fargate 프로파일 생성

다음 `create-fargate-profile` 예제에서는 포드를 시작할 IDs 서브넷과 함께 여러 네임스페이스 및 레이블이 있는 선택기에 대한 EKS Fargate 프로파일을 생성합니다.

```
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"}'
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 Amazon EKS 사용 설명서의 AWS Fargate 프로필 - Fargate 프로필 생성을 참조하세요.

예제 5: 포드를 시작할 서브넷과 함께 네임스페이스 및 레이블IDs에 대한 와일드카드 선택기를 사용하여 EKS Fargate 프로파일 생성

다음 `create-fargate-profile` 예제에서는 포드를 시작할 IDs 서브넷과 함께 여러 네임스페이스 및 레이블이 있는 선택기에 대한 EKS Fargate 프로파일을 생성합니다.

```
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"}'
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 Amazon EKS 사용 설명서의 AWS Fargate 프로필 - Fargate 프로필 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateFargateProfile의 섹션을 참조하세요. AWS CLI

### create-nodegroup

다음 코드 예시에서는 create-nodegroup을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: Amazon EKS 클러스터에 대한 관리형 노드 그룹을 생성합니다.

다음 create-nodegroup 예제에서는 Amazon EKS 클러스터에 대한 관리형 노드 그룹을 생성합니다.

```
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
```

출력:

```
{
    "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": {}
    }
```

```
}
```

자세한 내용은 Amazon EKS 사용 설명서의 관리형 노드 그룹 생성을 참조하세요.

예제 2: 사용자 지정 인스턴스 유형 및 디스크 크기를 사용하여 Amazon EKS 클러스터에 대한 관리형 노드 그룹을 생성합니다.

다음 `create-nodegroup` 예제에서는 사용자 지정 인스턴스 유형 및 디스크 크기가 있는 Amazon EKS 클러스터에 대한 관리형 노드 그룹을 생성합니다.

```
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
```

출력:

```
{
    "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": {}
    }
}
```

자세한 내용은 Amazon EKS 사용 설명서의 관리형 노드 그룹 생성을 참조하세요.

예제 3: 사용자 지정 인스턴스 유형, 디스크 크기, ami 유형, 용량 유형, 업데이트 구성, 레이블, taints 및 태그를 사용하여 Amazon EKS 클러스터에 대한 관리형 노드 그룹을 생성합니다.

다음 `create-nodegroup` 예제에서는 사용자 지정 인스턴스 유형, 디스크 크기, ami 유형, 용량 유형, 업데이트 구성, 레이블, 테인트 및 태그가 있는 Amazon EKS 클러스터에 대한 관리형 노드 그룹을 생성합니다.

```
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"}'
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 Amazon EKS 사용 설명서의 관리형 노드 그룹 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateNodegroup의 섹션을 참조하세요. AWS CLI

### `delete-addon`

다음 코드 예시에서는 `delete-addon`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1. Amazon EKS 추가 기능을 삭제하지만 EKS 클러스터에서 추가 기능 소프트웨어를 보존하려면

다음 `delete-addon` 예제 명령은 Amazon EKS 추가 기능을 삭제하지만 EKS 클러스터에서 추가 기능 소프트웨어를 유지합니다.

```
aws eks delete-addon \
    --cluster-name my-eks-cluster \
    --addon-name my-eks-addon \
    --preserve
```

출력:

```
{
    "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": {}
    }
}
```

자세한 내용은 Amazon EKS 추가 기능 관리 - Amazon 에서 추가 기능 삭제를 참조하세요. EKS

예 2. Amazon EKS 추가 기능을 삭제하고 EKS 클러스터에서 추가 기능 소프트웨어도 삭제하려면

다음 `delete-addon` 예제 명령은 Amazon EKS 추가 기능을 삭제하고 EKS 클러스터에서 추가 기능 소프트웨어도 삭제합니다.

```
aws eks delete-addon \
    --cluster-name my-eks-cluster \
    --addon-name my-eks-addon
```

출력:

```
{
    "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": {}
```

```
    }
}
```

자세한 내용은 Amazon EKS 추가 기능 관리 - Amazon 에서 추가 기능 삭제를 참조하세요.  EKS

- 자세한 API 내용은 명령 참조DeleteAddon의 섹션을 참조하세요. AWS CLI

### `delete-cluster`

다음 코드 예시에서는 `delete-cluster`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon EKS 클러스터 제어 영역 삭제

다음 `delete-cluster` 예제에서는 Amazon EKS 클러스터 제어 영역을 삭제합니다.

```
aws eks delete-cluster \
    --name my-eks-cluster
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon EKS 클러스터 삭제를 참조하세요. EKS

- 자세한 API 내용은 명령 참조DeleteCluster의 섹션을 참조하세요. AWS CLI

### `delete-fargate-profile`

다음 코드 예시에서는 `delete-fargate-profile`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 네임스페이스가 있는 선택기에 대한 EKS Fargate 프로파일 생성

다음 `delete-fargate-profile` 예제에서는 네임스페이스가 있는 선택기에 대한 EKS Fargate 프로파일을 생성합니다.

```
aws eks delete-fargate-profile \
    --cluster-name my-eks-cluster \
    --fargate-profile-name my-fargate-profile
```

출력:

```
{
    "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": {}
    }
}
```

자세한 내용은 Amazon EKS 사용 설명서의 AWS Fargate 프로파일 - Fargate 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteFargateProfile의 섹션을 참조하세요. AWS CLI

### `delete-nodegroup`

다음 코드 예시에서는 `delete-nodegroup`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: Amazon EKS 클러스터의 관리형 노드 그룹 삭제

다음 `delete-nodegroup` 예제에서는 Amazon EKS 클러스터의 관리형 노드 그룹을 삭제합니다.

```
aws eks delete-nodegroup \
    --cluster-name my-eks-cluster \
    --nodegroup-name my-eks-nodegroup
```

출력:

```
{
    "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"
        }
    }
}
```

- 자세한 API 내용은 명령 참조DeleteNodegroup의 섹션을 참조하세요. AWS CLI

### deregister-cluster

다음 코드 예시에서는 deregister-cluster을 사용하는 방법을 보여 줍니다.

AWS CLI

연결된 클러스터를 등록 취소하여 Amazon EKS 제어 영역에서 제거하려면

다음 deregister-cluster 예제에서는 연결된 클러스터를 등록 취소하여 Amazon EKS 제어 영역에서 제거합니다.

```
aws eks deregister-cluster \
    --name my-eks-anywhere-cluster
```

출력:

```
{
    "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/AmazonEKSConnectorAgentRole"
        }
    }
}
```

자세한 내용은 Amazon EKS 사용 설명서의 클러스터 등록 취소를 참조하세요.

- 자세한 API 내용은 명령 참조DeregisterCluster의 섹션을 참조하세요. AWS CLI

### describe-addon-configuration

다음 코드 예시에서는 describe-addon-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: Amazon vpc-cni를 생성하거나 업데이트할 때 사용할 수 있는 구성 옵션 AddOns

다음 describe-addon-configuration 예제에서는 vpc-cni 추가 기능에 대한 추가 기능이 생성되거나 해당 버전으로 업데이트될 때 사용하는 사용 가능한 모든 구성 스키마를 반환합니다.

```
aws eks describe-addon-configuration \
    --addon-name vpc-cni \
    --addon-version v1.15.1-eksbuild.1
```

출력:

```
{
    "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\"}"
}
```

예제 2: Amazon 코어를 생성하거나 업데이트할 때 사용할 수 있는 구성 옵션 AddOns

다음 describe-addon-configuration 예제에서는 코어 추가 기능에 대한 추가 기능이 생성되거나 업데이트될 때 사용하는 사용 가능한 모든 구성 스키마를 해당 버전으로 반환합니다.

```
aws eks describe-addon-configuration \
    --addon-name coredns \
    --addon-version v1.8.7-eksbuild.4
```

출력:

```
{
    "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\"}}}"
}
```

자세한 내용은 Amazon 에서 Amazon EKS 클러스터에 대한 kubeconfig 파일 생성 또는 업데이트를 참조하세요EKS.

- 자세한 API 내용은 명령 참조DescribeAddonConfiguration의 섹션을 참조하세요. AWS CLI

### describe-addon-versions

다음 코드 예시에서는 describe-addon-versions을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: EKS 클러스터에 사용할 수 있는 모든 추가 기능 나열

다음 describe-addon-versions 예제에서는 사용 가능한 AWS 모든 추가 기능을 나열합니다.

```
aws eks describe-addon-versions  \
    --query 'sort_by(addons  &owner)[].{publisher: publisher, owner: owner,
 addonName: addonName, type: type}' \
    --output table
```

출력:

```
----------------------------------------------------------------------------------------------
|                                                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       |
|  tetrate-io_istio-distro                     |  aws-marketplace |  tetrate-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               |
+-------------------------------------------------+------------------
+----------------------+-------------------------+
```

자세한 내용은 Amazon 사용 설명서의 Amazon EKS 추가 기능 관리 - 추가 기능 생성을 참조하세요. EKS

예제 2: 에 대해 제공된 지정된 Kubernetes 버전에 사용 가능한 모든 추가 기능 나열 EKS

다음 `describe-addon-versions` 예제에서는 에 대해 지원되는 지정된 Kubernetes 버전에 사용 가능한 모든 추가 기능을 나열합니다EKS.

```
aws eks describe-addon-versions  \
    --kubernetes-version=1.26 \
    --query 'sort_by(addons  &owner)[].{publisher: publisher, owner: owner,
 addonName: addonName, type: type}' \
    --output table
```

출력:

```
-----------------------------------------------------------------------------------------------------
|                                                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               |
|  tetrate-io_istio-distro                    |  aws-marketplace |  tetrate-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               |
+------------------------------------------------+------------------
+----------------------+-------------------------+
```

자세한 내용은 Amazon 사용 설명서의 Amazon EKS 추가 기능 관리 - 추가 기능 생성을 참조하세요. EKS

예제 3: 에 대해 지원되는 지정된 Kubernetes 버전에 대해 사용 가능한 모든 vpc-cni 추가 기능 버전을 나열합니다. EKS

다음 `describe-addon-versions` 예제에서는 에 대해 지원되는 지정된 Kubernetes 버전에 대해 사용 가능한 모든 vpc-cni 추가 기능 버전을 나열합니다EKS.

```
aws eks describe-addon-versions \
    --kubernetes-version=1.26 \
    --addon-name=vpc-cni \
    --query='addons[].addonVersions[].addonVersion'
```

출력:

```
[
    "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"
]
```

자세한 내용은 Amazon 사용 설명서의 Amazon EKS 추가 기능 관리 - 추가 기능 생성을 참조하세요. EKS

- 자세한 API 내용은 명령 참조DescribeAddonVersions의 섹션을 참조하세요. AWS CLI

### `describe-addon`

다음 코드 예시에서는 `describe-addon`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon EKS 클러스터에서 적극적으로 실행되는 EKS 추가 기능 설명

다음 `describe-addon` 예제는 Amazon EKS 클러스터에서 EKS 애드온을 적극적으로 실행하고 있습니다.

```
aws eks describe-addon \
    --cluster-name my-eks-cluster \
    --addon-name vpc-cni
```

출력:

```
{
    "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-YfakrqOC1UTm",
        "tags": {
            "eks-addon-key-3": "value-3",
            "eks-addon-key-4": "value-4"
        },
        "configurationValues": "resources:\n    limits:\n        cpu: '100m'\nenv:\n
    AWS_VPC_K8S_CNI_LOGLEVEL: 'DEBUG'"
    }
}
```

- 자세한 API 내용은 명령 참조DescribeAddon의 섹션을 참조하세요. AWS CLI

### `describe-cluster`

다음 코드 예시에서는 `describe-cluster`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon EKS 클러스터에서 적극적으로 실행되는 EKS 추가 기능 설명

다음 `describe-cluster` 예제는 Amazon EKS 클러스터에서 EKS 애드온을 적극적으로 실행합니다.

```
aws eks describe-cluster \
    --cluster-name my-eks-cluster
```

출력:

```
{
    "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.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": [
                "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"
        }
    }
}
```

- 자세한 API 내용은 명령 참조DescribeCluster의 섹션을 참조하세요. AWS CLI

### describe-fargate-profile

다음 코드 예시에서는 `describe-fargate-profile`을 사용하는 방법을 보여 줍니다.

AWS CLI

Fargate 프로필 설명

다음 `describe-fargate-profile` 예제에서는 Fargate 프로파일에 대해 설명합니다.

```
aws eks describe-fargate-profile \
    --cluster-name my-eks-cluster \
    --fargate-profile-name my-fargate-profile
```

출력:

```
{
    "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-1htfAaJdJUEO",
        "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"
        }
    }
}
```

- 자세한 API 내용은 명령 참조DescribeFargateProfile의 섹션을 참조하세요. AWS CLI

### describe-identity-provider-config

다음 코드 예시에서는 describe-identity-provider-config을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon EKS 클러스터와 연결된 자격 증명 공급자 구성 설명

다음 `describe-identity-provider-config` 예제에서는 Amazon EKS 클러스터와 연결된 자격 증명 공급자 구성을 설명합니다.

```
aws eks describe-identity-provider-config \
    --cluster-name my-eks-cluster \
    --identity-provider-config type=oidc,name=my-identity-provider
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 Amazon EKS 사용 설명서의 OpenID Connect 자격 증명 공급자에서 클러스터 사용자 인증을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeIdentityProviderConfig의 섹션을 참조하세요. AWS CLI

### `describe-nodegroup`

다음 코드 예시에서는 `describe-nodegroup`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon EKS 클러스터의 관리형 노드 그룹 설명

다음 `describe-nodegroup` 예제에서는 Amazon EKS 클러스터의 관리형 노드 그룹에 대해 설명합니다.

```
aws eks describe-nodegroup \
    --cluster-name my-eks-cluster \
    --nodegroup-name my-eks-nodegroup
```

출력:

```
{
    "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": {}
    }
}
```

- 자세한 API 내용은 명령 참조DescribeNodegroup의 섹션을 참조하세요. AWS CLI

### `describe-update`

다음 코드 예시에서는 `describe-update`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 클러스터에 대한 업데이트를 설명하려면

다음 `describe-update` 예제에서는 라는 클러스터에 대한 업데이트를 설명합니다.

```
aws eks describe-update \
    --name my-eks-cluster \
    --update-id 10bddb13-a71b-425a-b0a6-71cd03e59161
```

출력:

```
{
```

```
    "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": []
    }
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon EKS 클러스터 Kubernetes 버전 업데이트를 참조하세요. EKS

예제 2: 클러스터에 대한 업데이트를 설명하려면

다음 describe-update 예제에서는 라는 클러스터에 대한 업데이트를 설명합니다.

```
aws eks describe-update \
    --name my-eks-cluster \
    --update-id e4994991-4c0f-475a-a040-427e6da52966
```

출력:

```
{
    "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": []
    }
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon EKS 클러스터 Kubernetes 버전 업데이트를 참조하세요. EKS

예제 3: 클러스터에 대한 업데이트를 설명하려면

다음 describe-update 예제에서는 라는 클러스터에 대한 업데이트를 설명합니다.

```
aws eks describe-update \
    --name my-eks-cluster \
    --update-id b5f0ba18-9a87-4450-b5a0-825e6e84496f
```

출력:

```
{
    "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": []
    }
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon EKS 클러스터 Kubernetes 버전 업데이트를 참조하세요. EKS

- 자세한 API 내용은 명령 참조DescribeUpdate의 섹션을 참조하세요. AWS CLI

### disassociate-identity-provider-config

다음 코드 예시에서는 disassociate-identity-provider-config을 사용하는 방법을 보여 줍니다.

AWS CLI

자격 증명 공급자를 Amazon EKS 클러스터에 연결 해제

다음 disassociate-identity-provider-config 예제에서는 자격 증명 공급자를 Amazon EKS 클러스터에 연결 해제합니다.

```
aws eks disassociate-identity-provider-config \
    --cluster-name my-eks-cluster \
    --identity-provider-config 'type=oidc,name=my-identity-provider'
```

출력:

```
{
    "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": []
    }
}
```

자세한 내용은 Amazon EKS 사용 설명서의 OpenID Connect 자격 증명 공급자에서 클러스터 사용자 인증 - 클러스터에서 OIDC 자격 증명 공급자 연결 해제를 참조하세요.

- 자세한 API 내용은 명령 참조DisassociateIdentityProviderConfig의 섹션을 참조하세요. AWS CLI

### get-token

다음 코드 예시에서는 get-token을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: `my-eks-cluster`라는 Amazon EKS 클러스터의 인증 토큰 가져오기

다음 `get-token` 예제에서는 라는 Amazon EKS 클러스터의 인증 토큰을 가져옵니다 my-eks-cluster.

```
aws eks get-token \
    --cluster-name my-eks-cluster
```

출력:

```
{
    "kind": "ExecCredential",
    "apiVersion": "client.authentication.k8s.io/v1beta1",
    "spec": {},
    "status": {
        "expirationTimestamp": "2024-04-11T20:59:56Z",
        "token": "k8s-aws-v1.EXAMPLE_TOKEN_DATA_STRING..."
    }
}
```

예제 2: 토큰에 서명할 때 자격 증명에 대해 이 역할을ARN 수임하여 `my-eks-cluster`라는 Amazon EKS 클러스터의 인증 토큰을 가져옵니다.

다음 `get-token` 예제에서는 토큰에 서명할 때 자격 증명에 대해 이 역할을ARN 수임하여 라는 my-eks-cluster Amazon EKS 클러스터의 인증 토큰을 가져옵니다.

```
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
```

출력:

```
{
    "kind": "ExecCredential",
    "apiVersion": "client.authentication.k8s.io/v1beta1",
    "spec": {},
    "status": {
        "expirationTimestamp": "2024-04-11T21:05:26Z",
```

```
        "token": "k8s-aws-v1.EXAMPLE_TOKEN_DATA_STRING..."
    }
}
```

- 자세한 API 내용은 명령 참조GetToken의 섹션을 참조하세요. AWS CLI

### `list-addons`

다음 코드 예시에서는 `list-addons`을 사용하는 방법을 보여 줍니다.

AWS CLI

`my-eks-cluster`라는 Amazon EKS 클러스터에 설치된 모든 추가 기능 나열

다음 `list-addons` 예제에서는 라는 Amazon EKS 클러스터에 설치된 모든 추가 기능을 나열합니다 my-eks-cluster.

```
aws eks list-addons \
    --cluster-name my-eks-cluster
```

출력:

```
{
    "addons": [
        "kube-proxy",
        "vpc-cni"
    ]
}
```

- 자세한 API 내용은 명령 참조ListAddons의 섹션을 참조하세요. AWS CLI

### `list-clusters`

다음 코드 예시에서는 `list-clusters`을 사용하는 방법을 보여 줍니다.

AWS CLI

`my-eks-cluster`라는 Amazon EKS 클러스터에 설치된 모든 추가 기능을 나열하려면

다음 `list-clusters` 예제에서는 라는 Amazon EKS 클러스터에 설치된 모든 추가 기능을 나열합니다 my-eks-cluster.

```
aws eks list-clusters
```

출력:

```
{
    "clusters": [
        "prod",
        "qa",
        "stage",
        "my-eks-cluster"
    ]
}
```

- 자세한 API 내용은 명령 참조ListClusters의 섹션을 참조하세요. AWS CLI

### `list-fargate-profiles`

다음 코드 예시에서는 `list-fargate-profiles`을 사용하는 방법을 보여 줍니다.

AWS CLI

`my-eks-cluster`라는 Amazon EKS 클러스터의 모든 fargate 프로파일을 나열하려면

다음 `list-fargate-profiles` 예제에서는 라는 Amazon EKS 클러스터의 모든 fargate 프로파일을 나열합니다 my-eks-cluster.

```
aws eks list-fargate-profiles \
    --cluster-name my-eks-cluster
```

출력:

```
{
    "fargateProfileNames": [
        "my-fargate-profile"
    ]
}
```

- 자세한 API 내용은 명령 참조ListFargateProfiles의 섹션을 참조하세요. AWS CLI

### `list-identity-provider-configs`

다음 코드 예시에서는 `list-identity-provider-configs`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon EKS 클러스터에 연결된 자격 증명 공급자 나열

다음 `list-identity-provider-configs` 예제에서는 Amazon EKS 클러스터와 연결된 자격 증명 공급자를 나열합니다.

```
aws eks list-identity-provider-configs \
    --cluster-name my-eks-cluster
```

출력:

```
{
    "identityProviderConfigs": [
        {
            "type": "oidc",
            "name": "my-identity-provider"
        }
    ]
}
```

자세한 내용은 Amazon EKS 사용 설명서의 OpenID Connect 자격 증명 공급자에서 클러스터 사용자 인증을 참조하세요.

- 자세한 API 내용은 명령 참조ListIdentityProviderConfigs의 섹션을 참조하세요. AWS CLI

### `list-nodegroups`

다음 코드 예시에서는 `list-nodegroups`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon EKS 클러스터의 모든 노드 그룹 나열

다음 `list-nodegroups` 예제에서는 Amazon EKS 클러스터의 모든 노드 그룹을 나열합니다.

```
aws eks list-nodegroups \
```

```
    --cluster-name my-eks-cluster
```

출력:

```
{
    "nodegroups": [
        "my-eks-managed-node-group",
        "my-eks-nodegroup"
    ]
}
```

- 자세한 API 내용은 명령 참조ListNodegroups의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: Amazon EKS 클러스터의 모든 태그를 나열하려면 ARN

다음 `list-tags-for-resource` 예제에서는 Amazon EKS 클러스터 의 모든 태그를 나열합니다 ARN.

```
aws eks list-tags-for-resource \
    --resource-arn arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster
```

출력:

```
{
    "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"
    }
}
```

예제 2: Amazon EKS 노드 그룹의 모든 태그를 나열하려면 ARN

다음 `list-tags-for-resource` 예제에서는 Amazon EKS 노드 그룹 에 대한 모든 태그를 나열합니다ARN.

```
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
```

출력:

```
{
    "tags": {
        "aws:cloudformation:stack-name": "eksctl-my-eks-cluster-nodegroup-my-eks-
managed-node-group",
        "aws:cloudformation:stack-id": "arn:aws:cloudformation:us-
east-2:111122223333:stack/eksctl-my-eks-cluster-nodegroup-my-eks-managed-node-group/
eaa20310-e219-11ee-b851-0ab9ad8228ff",
        "eksctl.cluster.k8s.io/v1alpha1/cluster-name": "my-eks-cluster",
        "EKS-Cluster-Name": "my-eks-cluster",
        "alpha.eksctl.io/nodegroup-type": "managed",
        "NodeGroup Name 1": "my-eks-managed-node-group",
        "k8s.io/cluster-autoscaler/enabled": "true",
        "nodegroup-role": "worker",
        "alpha.eksctl.io/cluster-name": "my-eks-cluster",
        "alpha.eksctl.io/nodegroup-name": "my-eks-managed-node-group",
        "karpenter.sh/discovery": "my-eks-cluster",
        "NodeGroup Name 2": "AmazonLinux-Linux-Managed-NG-v1-26-v1",
        "auto-delete": "no",
        "k8s.io/cluster-autoscaler/my-eks-cluster": "owned",
        "aws:cloudformation:logical-id": "ManagedNodeGroup",
        "alpha.eksctl.io/eksctl-version": "0.173.0-dev
+a7ee89342.2024-03-01T03:40:57Z"
    }
```

```
}
```

예제 3: Amazon EKS Fargate 프로필의 모든 태그를 나열하려면 ARNe

다음 `list-tags-for-resource` 예제에서는 Amazon EKS Fargate 프로파일 의 모든 태그를 나열합니다ARN.

```
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
```

출력:

```
{
    "tags": {
        "eks-fargate-profile-key-2": "value-2",
        "eks-fargate-profile-key-1": "value-1"
    }
}
```

예제 4: Amazon EKS 추가 기능의 모든 태그를 나열하려면 ARN

다음 `list-tags-for-resource` 예제에서는 Amazon EKS 추가 기능 에 대한 모든 태그를 나열합니다ARN.

```
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
```

출력:

```
{
    "tags": {
        "eks-addon-key-2": "value-2",
        "eks-addon-key-1": "value-1"
    }
}
```

예제 5: Amazon EKS OIDC 자격 증명 공급자의 모든 태그를 나열하려면 ARN

다음 `list-tags-for-resource` 예제에서는 Amazon EKS OIDC 자격 증명 공급자 의 모든 태그를 나열합니다ARN.

```
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
```

출력:

```
{
    "tags": {
        "my-identity-provider": "test"
    }
}
```

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### `list-update`

다음 코드 예시에서는 `list-update`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: Amazon EKS 클러스터 이름과 연결된 업데이트를 나열하려면

다음 `list-updates` 예제에서는 Amazon EKS 클러스터 이름에 IDs 대한 모든 업데이트를 나열합니다.

```
aws eks list-updates \
    --name my-eks-cluster
```

출력:

```
{
    "updateIds": [
        "5f78d14e-c57b-4857-a3e4-cf664ae20949",
        "760e5a3f-adad-48c7-88d3-7ac283c09c26",
        "cd4ec863-bc55-47d5-a377-3971502f529b",
        "f12657ce-e869-4f17-b158-a82ab8b7d937"
    ]
}
```

예제 2: Amazon EKS 노드 그룹에 IDs 대한 모든 업데이트를 나열하려면

다음 `list-updates` 예제에서는 Amazon EKS 노드 그룹에 IDs 대한 모든 업데이트를 나열합니다.

```
aws eks list-updates \
    --name my-eks-cluster \
    --nodegroup-name my-eks-managed-node-group
```

출력:

```
{
    "updateIds": [
        "8c6c1bef-61fe-42ac-a242-89412387b8e7"
    ]
}
```

예제 3: Amazon EKS Add-one의 모든 업데이트를 나열IDs하려면

다음 `list-updates` 예제에서는 Amazon EKS 추가 기능에 IDs 대한 모든 업데이트를 나열합니다.

```
aws eks list-updates \
    --name my-eks-cluster \
    --addon-name vpc-cni
```

출력:

```
{
    "updateIds": [
        "9cdba8d4-79fb-3c83-afe8-00b508d33268"
    ]
}
```

- 자세한 API 내용은 명령 참조ListUpdate의 섹션을 참조하세요. AWS CLI

## list-updates

다음 코드 예시에서는 `list-updates`을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터에 대한 업데이트를 나열하려면

이 예제 명령은 기본 리전example에 이름이 지정된 클러스터의 현재 업데이트를 나열합니다.

명령:

```
aws eks list-updates --name example
```

출력:

```
{
    "updateIds": [
        "10bddb13-a71b-425a-b0a6-71cd03e59161"
    ]
}
```

- 자세한 API 내용은 명령 참조ListUpdates의 섹션을 참조하세요. AWS CLI

### register-cluster

다음 코드 예시에서는 register-cluster을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: Amazon에 외부 EKS_ANYWHERE Kubernetes 클러스터 등록 EKS

다음 register-cluster 예제에서는 외부 EKS_ANYWHERE Kubernetes 클러스터를 Amazon에 등록합니다EKS.

```
aws eks register-cluster \
    --name my-eks-anywhere-cluster \
    --connector-config 'roleArn=arn:aws:iam::111122223333:role/
AmazonEKSConnectorAgentRole,provider=EKS_ANYWHERE'
```

출력:

```
{
    "cluster": {
```

```
        "name": "my-eks-anywhere-cluster",
        "arn": "arn:aws:eks:us-east-2:111122223333:cluster/my-eks-anywhere-cluster",
        "createdAt": "2024-04-12T12:38:37.561000-04:00",
        "status": "PENDING",
        "tags": {},
        "connectorConfig": {
            "activationId": "xxxxxxxxACTIVATION_IDxxxxxxxx",
            "activationCode": "xxxxxxxxACTIVATION_CODExxxxxxxx",
            "activationExpiry": "2024-04-15T12:38:37.082000-04:00",
            "provider": "EKS_ANYWHERE",
            "roleArn": "arn:aws:iam::111122223333:role/AmazonEKSConnectorAgentRole"
        }
    }
}
```

자세한 내용은 Amazon EKS 사용 설명서의 외부 클러스터 연결을 참조하세요.

예제 2: Amazon에 외부 Kubernetes 클러스터 등록 EKS

다음 `register-cluster` 예제에서는 외부 EKS_ANYWHERE Kubernetes 클러스터를 Amazon에 등록합니다EKS.

```
aws eks register-cluster \
    --name my-eks-anywhere-cluster \
    --connector-config 'roleArn=arn:aws:iam::111122223333:role/
AmazonEKSConnectorAgentRole,provider=OTHER'
```

출력:

```
{
    "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/AmazonEKSConnectorAgentRole"
```

```
        }
    }
}
```

자세한 내용은 Amazon EKS 사용 설명서의 외부 클러스터 연결을 참조하세요.

- 자세한 API 내용은 명령 참조RegisterCluster의 섹션을 참조하세요. AWS CLI

### tag-resource

다음 코드 예시에서는 tag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: Amazon EKS 클러스터에 지정된 태그를 추가하려면

다음 tag-resource 예제에서는 지정된 태그를 Amazon EKS 클러스터에 추가합니다.

```
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'
```

이 명령은 출력을 생성하지 않습니다.

예제 2: Amazon EKS 노드 그룹에 지정된 태그를 추가하려면

다음 tag-resource 예제에서는 지정된 태그를 Amazon EKS 노드 그룹에 추가합니다.

```
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'
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### untag-resource

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: Amazon EKS 클러스터에서 지정된 태그를 삭제하는 방법

다음 `untag-resource` 예제에서는 Amazon EKS 클러스터에서 지정된 태그를 삭제합니다.

```
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"
```

이 명령은 출력을 생성하지 않습니다.

예제 2: Amazon EKS 노드 그룹에서 지정된 태그를 삭제하는 방법

다음 `untag-resource` 예제에서는 Amazon EKS 노드 그룹에서 지정된 태그를 삭제합니다.

```
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"
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

## update-addon

다음 코드 예시에서는 `update-addon`을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1. 서비스 계정 역할로 Amazon EKS 추가 기능을 업데이트하려면 ARN

다음 `update-addon` 예제 명령은 서비스 계정 역할 로 Amazon EKS 추가 기능을 업데이트합니다 ARN.

```
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-YfakrqOC1UTm
```

출력:

```
{
    "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-YfakrqOC1UTm"
            }
        ],
        "updatedAt": "2024-04-12T16:04:55.614000-04:00",
        "errors": []
    }
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon EKS 추가 기능 관리 - 추가 기능 업데이트를 참조하세요. EKS

예 2. 특정 EKS 추가 기능 버전으로 Amazon 추가 기능을 업데이트하려면

다음 update-addon 예제 명령은 특정 EKS 추가 기능 버전으로 Amazon 추가 기능을 업데이트합니다.

```
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-YfakrqOC1UTm \
    --addon-version v1.16.4-eksbuild.2
```

출력:

```
{
    "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-YfakrqOC1UTm"
            }
        ],
        "createdAt": "2024-04-12T16:07:16.550000-04:00",
        "errors": []
    }
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon EKS 추가 기능 관리 - 추가 기능 업데이트를 참조하세요. EKS

예제 3. 사용자 지정 구성 값으로 Amazon EKS 추가 기능을 업데이트하고 충돌 세부 정보를 해결하려면

다음 update-addon 예제 명령은 사용자 지정 구성 값으로 Amazon EKS 추가 기능을 업데이트하고 충돌 세부 정보를 해결합니다.

```
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-YfakrqOC1UTm \
    --addon-version v1.16.4-eksbuild.2 \
    --configuration-values '{"resources": {"limits":{"cpu":"100m"}, "requests":
{"cpu":"50m"}}}' \
    --resolve-conflicts PRESERVE
```

출력:

```
{
    "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-YfakrqOC1UTm"
            },
            {
                "type": "ResolveConflicts",
                "value": "PRESERVE"
            },
            {
                "type": "ConfigurationValues",
                "value": "{\"resources\": {\"limits\":{\"cpu\":\"100m\"}, \"requests
\":{\"cpu\":\"50m\"}}}"
            }
        ],
        "createdAt": "2024-04-12T16:16:27.363000-04:00",
        "errors": []
    }
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon EKS 추가 기능 관리 - 추가 기능 업데이트를 참조하세요. EKS

예제 4. 사용자 지정 JSON 구성 값 파일로 Amazon EKS 추가 기능을 업데이트하려면

다음 update-addon 예제 명령은 사용자 지정 JSON 구성 값으로 Amazon EKS 추가 기능을 업데이트하고 충돌 세부 정보를 해결합니다.

```
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-YfakrqOC1UTm \
    --addon-version v1.17.1-eksbuild.1 \
    --configuration-values 'file://configuration-values.json' \
    --resolve-conflicts PRESERVE
```

configuration-values.json의 콘텐츠:

```
{
    "resources": {
        "limits": {
            "cpu": "100m"
        },
        "requests": {
            "cpu": "50m"
        }
    },
    "env": {
        "AWS_VPC_K8S_CNI_LOGLEVEL": "ERROR"
    }
}
```

출력:

```
{
    "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-YfakrqOC1UTm"
            },
            {
                "type": "ResolveConflicts",
                "value": "PRESERVE"
            },
            {
                "type": "ConfigurationValues",
                "value": "{\n    \"resources\": {\n        \"limits\": {\n
  \"cpu\": \"100m\"\n        },\n        \"requests\": {\n            \"cpu\": \"50m
\"\n        }\n    },\n    \"env\": {\n        \"AWS_VPC_K8S_CNI_LOGLEVEL\": \"ERROR
\"\n    }\n}"
            }
        ],
```

```
        "createdAt": "2024-04-12T16:22:55.519000-04:00",
        "errors": []
    }
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon EKS 추가 기능 관리 - 추가 기능 업데이트를 참조하세요. EKS

예제 5. 사용자 지정 YAML 구성 값 파일로 Amazon EKS 추가 기능을 업데이트하려면

다음 update-addon 예제 명령은 사용자 지정 YAML 구성 값으로 Amazon EKS 추가 기능을 업데이트하고 충돌 세부 정보를 해결합니다.

```
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-YfakrqOC1UTm \
    --addon-version v1.18.0-eksbuild.1 \
    --configuration-values 'file://configuration-values.yaml' \
    --resolve-conflicts PRESERVE
```

configuration-values.yaml의 콘텐츠:

```
resources:
    limits:
        cpu: '100m'
    requests:
        cpu: '50m'
env:
    AWS_VPC_K8S_CNI_LOGLEVEL: 'DEBUG'
```

출력:

```
{
    "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-YfakrqOC1UTm"
            },
            {
                "type": "ResolveConflicts",
                "value": "PRESERVE"
            },
            {
                "type": "ConfigurationValues",
                "value": "resources:\n    limits:\n        cpu: '100m'\n
 requests:\n        cpu: '50m'\nenv:\n    AWS_VPC_K8S_CNI_LOGLEVEL: 'DEBUG'"
            }
        ],
        "createdAt": "2024-04-12T16:25:07.212000-04:00",
        "errors": []
    }
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon EKS 추가 기능 관리 - 추가 기능 업데이트를 참조하세요. EKS

- 자세한 API 내용은 명령 참조UpdateAddon의 섹션을 참조하세요. AWS CLI

### update-cluster-config

다음 코드 예시에서는 update-cluster-config을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터 엔드포인트 액세스를 업데이트하려면

이 예제 명령은 클러스터를 업데이트하여 엔드포인트 퍼블릭 액세스를 비활성화하고 프라이빗 엔드포인트 액세스를 활성화합니다.

명령:

```
aws eks update-cluster-config --name example \
```

```
--resources-vpc-config endpointPublicAccess=false,endpointPrivateAccess=true
```

출력:

```
{
    "update": {
        "id": "ec883c93-2e9e-407c-a22f-8f6fa6e67d4f",
        "status": "InProgress",
        "type": "EndpointAccessUpdate",
        "params": [
            {
                "type": "EndpointPublicAccess",
                "value": "false"
            },
            {
                "type": "EndpointPrivateAccess",
                "value": "true"
            }
        ],
        "createdAt": 1565806986.506,
        "errors": []
    }
}
```

클러스터에 대한 로깅을 활성화하려면

이 예제 명령은 라는 클러스터에 대한 모든 클러스터 제어 영역 로깅 유형을 활성화합니다`example`.

명령:

```
aws eks update-cluster-config --name example \
--logging '{"clusterLogging":[{"types":
["api","audit","authenticator","controllerManager","scheduler"],"enabled":true}]}'
```

출력:

```
{
    "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": []
    }
}
```

- 자세한 API 내용은 명령 참조UpdateClusterConfig의 섹션을 참조하세요. AWS CLI

### update-cluster-version

다음 코드 예시에서는 update-cluster-version을 사용하는 방법을 보여 줍니다.

AWS CLI

`my-eks-cluster`라는 Amazon EKS 클러스터를 지정된 Kubernetes 버전으로 업데이트하려면

다음 update-cluster-version 예제에서는 Amazon EKS 클러스터를 지정된 Kubernetes 버전으로 업데이트합니다.

```
aws eks update-cluster-version \
    --name my-eks-cluster \
    --kubernetes-version 1.27
```

출력:

```
{
    "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": []
    }
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon EKS 클러스터 Kubernetes 버전 업데이트를 참조하세요. EKS

- 자세한 API 내용은 명령 참조UpdateClusterVersion의 섹션을 참조하세요. AWS CLI

### update-kubeconfig

다음 코드 예시에서는 update-kubeconfig을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 'my-eks-cluster`라는 Amazon EKS 클러스터에 연결할 수 있도록 kubeconfig를 생성하거나 업데이트하여 kubectl을 구성합니다.

다음 update-kubeconfig 예제에서는 라는 Amazon EKS 클러스터에 연결할 수 있도록 kubeconfig를 생성하거나 업데이트하여 kubectl을 구성합니다 my-eks-cluster.

```
aws eks update-kubeconfig \
    --name my-eks-cluster
```

출력:

```
Updated context arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster in /Users/
xxx/.kube/config
```

자세한 내용은 Amazon 사용 설명서의 Amazon EKS 클러스터용 kubeconfig 파일 생성 또는 업데이트를 참조하세요. EKS

예제 2: 'my-eks-cluster`라는 Amazon EKS 클러스터에 연결할 수 있도록 kubeconfig(클러스터 인증을 위한 역할을 수임하는 role-arn 옵션 포함)를 생성하거나 업데이트하여 kubectl을 구성합니다.

다음 update-kubeconfig 예제에서는 라는 Amazon EKS 클러스터에 연결할 수 있도록 kubeconfig(클러스터 인증을 위한 역할을 수임하는 역할 획득 옵션 포함)를 생성하거나 업데이트하여 kubectl을 구성합니다 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
```

출력:

```
Updated context arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster in /Users/xxx/.kube/config
```

자세한 내용은 Amazon 사용 설명서의 Amazon EKS 클러스터에 대한 kubeconfig 파일 생성 또는 업데이트를 참조하세요. EKS

예제 3: 'my-eks-cluster`라는 Amazon EKS 클러스터에 연결할 수 있도록 kubeconfig(사용자 지정 클러스터 별칭 및 사용자 별칭과 함께 클러스터 인증을 위한 역할을 수임하는 role-arn 옵션 포함)를 생성하거나 업데이트하여 kubectl을 구성합니다.

다음 update-kubeconfig 예제에서는 라는 Amazon EKS 클러스터에 연결할 수 있도록 kubeconfig(사용자 지정 클러스터 별칭 및 사용자 별칭과 함께 클러스터 인증을 위한 역할을 수임하는 역할 획득 옵션 포함)를 생성하거나 업데이트하여 kubectl을 구성합니다 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
```

출력:

```
Updated context stage-eks-cluster in /Users/dubaria/.kube/config
```

자세한 내용은 Amazon 사용 설명서의 Amazon EKS 클러스터용 kubeconfig 파일 생성 또는 업데이트를 참조하세요. EKS

예제 4: 검토할 kubeconfig 파일 항목 인쇄 및 `my-eks-cluster`라는 Amazon EKS 클러스터에 연결할 수 있도록 kubectl 구성

다음 update-kubeconfig 예제에서는 라는 Amazon EKS 클러스터에 연결할 수 있도록 kubeconfig(사용자 지정 클러스터 별칭 및 사용자 별칭과 함께 클러스터 인증을 위한 역할을 수임하는 역할 획득 옵션 포함)를 생성하거나 업데이트하여 kubectl을 구성합니다 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
```

출력:

```
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.yl4.us-east-2.eks.amazonaws.com
name: arn:aws:eks:us-east-2:111122223333:cluster/my-eks-cluster
```

자세한 내용은 Amazon 사용 설명서의 Amazon EKS 클러스터에 대한 kubeconfig 파일 생성 또는 업데이트를 참조하세요.  EKS

- 자세한 API 내용은 명령 참조UpdateKubeconfig의 섹션을 참조하세요. AWS CLI

### update-nodegroup-config

다음 코드 예시에서는 update-nodegroup-config을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: 관리형 노드 그룹을 업데이트하여 Amazon EKS 클러스터의 EKS 작업자 노드에 새 레이블 및 taint 추가

다음 update-nodegroup-config 예제에서는 관리형 노드 그룹을 업데이트하여 Amazon EKS 클러스터의 EKS 작업자 노드에 새 레이블과 taint을 추가합니다.

```
aws eks update-nodegroup-config \
    --cluster-name my-eks-cluster \
    --nodegroup-name my-eks-nodegroup \
    --labels 'addOrUpdateLabels={my-eks-nodegroup-label-1=value-1,my-eks-nodegroup-label-2=value-2}' \
    --taints 'addOrUpdateTaints=[{key=taint-key-1,value=taint-value-1,effect=NO_EXECUTE}]'
```

출력:

```
{
    "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": []
    }
}
```

자세한 내용은 Amazon EKS 사용 설명서의 관리형 노드 그룹 업데이트를 참조하세요.

예제 2: Amazon EKS 클러스터의 EKS 작업자 노드에 대한 레이블 및 taint을 제거하도록 관리형 노드 그룹 업데이트

다음 `update-nodegroup-config` 예제에서는 관리형 노드 그룹을 업데이트하여 Amazon EKS 클러스터의 EKS 작업자 노드에 대한 레이블과 taint을 제거합니다.

```
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}]'
```

출력:

```
{
    "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": []
    }
}
```

자세한 내용은 Amazon EKS 사용 설명서의 관리형 노드 그룹 업데이트를 참조하세요.

예제 3: Amazon EKS 클러스터의 EKS 작업자 노드에 대한 레이블 및 taint을 제거하고 추가하도록 관리형 노드 그룹 업데이트

다음 `update-nodegroup-config` 예제에서는 관리형 노드 그룹을 업데이트하여 Amazon EKS 클러스터의 EKS 작업자 노드에 대한 레이블과 taint을 제거하고 추가합니다.

```
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}]'
```

출력:

```
{
    "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": []
    }
}
```

자세한 내용은 Amazon EKS 사용 설명서의 관리형 노드 그룹 업데이트를 참조하세요.

예제 4: 관리형 노드 그룹을 업데이트하여 Amazon EKS 클러스터의 EKS 작업자 노드에 대한 Scaling-config 및 update-config를 업데이트합니다.

다음 `update-nodegroup-config` 예제에서는 관리형 노드 그룹을 업데이트하여 Amazon EKS 클러스터의 EKS 작업자 노드에 대한 Scaling-config 및 update-config를 업데이트합니다.

```
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
```

출력:

```
{
    "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": []
    }
}
```

자세한 내용은 Amazon EKS 사용 설명서의 관리형 노드 그룹 업데이트를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateNodegroupConfig의 섹션을 참조하세요. AWS CLI

### update-nodegroup-version

다음 코드 예시에서는 update-nodegroup-version을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: Amazon EKS 관리형 노드 그룹의 Kubernetes 버전 또는 AMI 버전 업데이트

다음 update-nodegroup-version 예제에서는 Amazon EKS 관리형 노드 그룹의 Kubernetes 버전 또는 AMI 버전을 Kubernetes 클러스터에 사용 가능한 최신 버전으로 업데이트합니다.

```
aws eks update-nodegroup-version \
    --cluster-name my-eks-cluster \
    --nodegroup-name my-eks-nodegroup \
    --no-force
```

출력:

```
{
    "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": []
    }
}
```

자세한 내용은 Amazon EKS 사용 설명서의 관리형 노드 그룹 업데이트를 참조하세요.

예제 2: Amazon EKS 관리형 노드 그룹의 Kubernetes 버전 또는 AMI 버전 업데이트

다음 `update-nodegroup-version` 예제에서는 Amazon EKS 관리형 노드 그룹의 Kubernetes 버전 또는 AMI 버전을 지정된 AMI 릴리스 버전으로 업데이트합니다.

```
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
```

출력:

```
{
    "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": []
    }
}
```

자세한 내용은 Amazon EKS 사용 설명서의 관리형 노드 그룹 업데이트 - <https://docs.aws.amazon.com/eks/latest/userguide/update-managed-node-group.html>`를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateNodegroupVersion의 섹션을 참조하세요. AWS CLI

## 를 사용한 Elastic Beanstalk 예제 AWS CLI

다음 코드 예제에서는 Elastic Beanstalk와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `abort-environment-update`

다음 코드 예시에서는 `abort-environment-update`을 사용하는 방법을 보여 줍니다.

AWS CLI

배포를 중단하려면

다음 명령은 라는 환경에 대해 실행 중인 애플리케이션 버전 배포를 중단합니다my-env.

```
aws elasticbeanstalk abort-environment-update --environment-name my-env
```

- 자세한 API 내용은 명령 참조AbortEnvironmentUpdate의 섹션을 참조하세요. AWS CLI

### check-dns-availability

다음 코드 예시에서는 check-dns-availability을 사용하는 방법을 보여 줍니다.

AWS CLI

의 가용성을 확인하려면 CNAME

다음 명령은 하위 도메인 의 가용성을 확인합니다my-cname.elasticbeanstalk.com.

```
aws elasticbeanstalk check-dns-availability --cname-prefix my-cname
```

출력:

```
{
    "Available": true,
    "FullyQualifiedCNAME": "my-cname.elasticbeanstalk.com"
}
```

- 자세한 API 내용은 명령 참조CheckDnsAvailability의 섹션을 참조하세요. AWS CLI

### create-application-version

다음 코드 예시에서는 create-application-version을 사용하는 방법을 보여 줍니다.

AWS CLI

새 애플리케이션 버전을 생성하려면

다음 명령은 새 버전인 “v1”의 애플리케이션을 생성합니다MyApp.

```
aws elasticbeanstalk create-application-version --application-name MyApp
 --version-label v1 --description MyAppv1 --source-bundle S3Bucket="my-
bucket",S3Key="sample.war" --auto-create-application
```

옵션 때문에 auto-create-application 애플리케이션이 아직 존재하지 않는 경우 자동으로 생성됩니다. 소스 번들은 Apache Tomcat 샘플 애플리케이션이 포함된 'my-bucket'이라는 s3 버킷에 저장된 .war 파일입니다.

출력:

```
{
  "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"
      }
  }
}
```

- 자세한 API 내용은 명령 참조CreateApplicationVersion의 섹션을 참조하세요. AWS CLI

### `create-application`

다음 코드 예시에서는 `create-application`을 사용하는 방법을 보여 줍니다.

AWS CLI

새 애플리케이션을 생성하려면

다음 명령은 “MyApp”라는 새 애플리케이션을 생성합니다.

```
aws elasticbeanstalk create-application --application-name MyApp --description "my
 application"
```

`create-application` 명령은 애플리케이션의 이름과 설명만 구성합니다. 애플리케이션의 소스 코드를 업로드하려면 를 사용하여 애플리케이션의 초기 버전을 생성합니다`create-`

application-version. 에는 애플리케이션과 애플리케이션 버전을 한 번에 생성할 수 있는 auto-create-application 옵션create-application-version도 있습니다.

출력:

```
{
  "Application": {
      "ApplicationName": "MyApp",
      "ConfigurationTemplates": [],
      "DateUpdated": "2015-02-12T18:32:21.181Z",
      "Description": "my application",
      "DateCreated": "2015-02-12T18:32:21.181Z"
  }
}
```

- 자세한 API 내용은 명령 참조CreateApplication의 섹션을 참조하세요. AWS CLI

### create-configuration-template

다음 코드 예시에서는 create-configuration-template을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 템플릿을 생성하려면

다음 명령은 id 를 사용하여 환경에 적용된 설정my-app-v1에서 라는 구성 템플릿을 생성합니다e-rpqsewtp2j.

```
aws elasticbeanstalk create-configuration-template --application-name my-app --
template-name my-app-v1 --environment-id e-rpqsewtp2j
```

출력:

```
{
    "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"
}
```

- 자세한 API 내용은 명령 참조CreateConfigurationTemplate의 섹션을 참조하세요. AWS CLI

### create-environment

다음 코드 예시에서는 create-environment을 사용하는 방법을 보여 줍니다.

AWS CLI

애플리케이션에 대한 새 환경을 생성하려면

다음 명령은 'my-app'이라는 java 애플리케이션의 버전 'v1'에 대한 새 환경을 생성합니다.

```
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"
```

출력:

```
{
  "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 는 이전에 로 업로드된 애플리케이션 버전의 레이블입니다 create-application-version.

환경 구성 옵션을 정의할 JSON 파일을 지정하려면

다음 create-environment 명령은 이름이 인 JSON 파일을 myoptions.json 사용하여 솔루션 스택 또는 구성 템플릿에서 얻은 값을 재정의하도록 지정합니다.

```
aws elasticbeanstalk create-environment --environment-name sample-env --application-
name sampleapp --option-settings file://myoptions.json
```

myoptions.json 는 여러 설정을 정의하는 JSON 객체입니다.

```
[
  {
    "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"
  }
]
```

자세한 내용은 AWS Elastic Beanstalk 개발자 안내서의 옵션 값을 참조하세요.

- 자세한 API 내용은 명령 참조CreateEnvironment의 섹션을 참조하세요. AWS CLI

### create-storage-location

다음 코드 예시에서는 create-storage-location을 사용하는 방법을 보여 줍니다.

AWS CLI

스토리지 위치를 생성하려면

다음 명령은 Amazon S3에 스토리지 위치를 생성합니다.

```
aws elasticbeanstalk create-storage-location
```

출력:

```
{
    "S3Bucket": "elasticbeanstalk-us-west-2-0123456789012"
}
```

- 자세한 API 내용은 명령 참조CreateStorageLocation의 섹션을 참조하세요. AWS CLI

### delete-application-version

다음 코드 예시에서는 delete-application-version을 사용하는 방법을 보여 줍니다.

AWS CLI

애플리케이션 버전을 삭제하려면

다음 명령은 라는 애플리케이션에 22a0-stage-150819_182129 대해 라는 애플리케이션 버전을 삭제합니다my-app.

```
aws elasticbeanstalk delete-application-version --version-label 22a0-stage-150819_182129 --application-name my-app
```

- 자세한 API 내용은 명령 참조DeleteApplicationVersion의 섹션을 참조하세요. AWS CLI

### delete-application

다음 코드 예시에서는 delete-application을 사용하는 방법을 보여 줍니다.

AWS CLI

애플리케이션 삭제

다음 명령은 라는 애플리케이션을 삭제합니다my-app.

```
aws elasticbeanstalk delete-application --application-name my-app
```

- 자세한 API 내용은 명령 참조DeleteApplication의 섹션을 참조하세요. AWS CLI

### delete-configuration-template

다음 코드 예시에서는 delete-configuration-template을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 템플릿을 삭제하려면

다음 명령은 라는 애플리케이션에 my-template 대한 라는 구성 템플릿을 삭제합니다my-app.

```
aws elasticbeanstalk delete-configuration-template --template-name my-template --
application-name my-app
```

- 자세한 API 내용은 명령 참조DeleteConfigurationTemplate의 섹션을 참조하세요. AWS CLI

### delete-environment-configuration

다음 코드 예시에서는 delete-environment-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

초안 구성을 삭제하려면

다음 명령은 라는 환경에 대한 초안 구성을 삭제합니다my-env.

```
aws elasticbeanstalk delete-environment-configuration --environment-name my-env --
application-name my-app
```

- 자세한 API 내용은 명령 참조DeleteEnvironmentConfiguration의 섹션을 참조하세요. AWS CLI

### describe-application-versions

다음 코드 예시에서는 describe-application-versions을 사용하는 방법을 보여 줍니다.

AWS CLI

애플리케이션 버전에 대한 정보를 보려면

다음 명령은 레이블이 지정된 애플리케이션 버전에 대한 정보를 검색합니다. v2

```
aws elasticbeanstalk describe-application-versions --application-name my-app --
version-label "v2"
```

출력:

```
{
    "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"
          }
      }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeApplicationVersions의 섹션을 참조하세요. AWS CLI

### `describe-applications`

다음 코드 예시에서는 `describe-applications`을 사용하는 방법을 보여 줍니다.

AWS CLI

애플리케이션 목록을 보려면

다음 명령은 현재 리전의 애플리케이션에 대한 정보를 검색합니다.

```
aws elasticbeanstalk describe-applications
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeApplications의 섹션을 참조하세요. AWS CLI

### `describe-configuration-options`

다음 코드 예시에서는 `describe-configuration-options`을 사용하는 방법을 보여 줍니다.

AWS CLI

환경에 대한 구성 옵션을 보려면

다음 명령은 라는 환경에 사용 가능한 모든 구성 옵션에 대한 설명을 검색합니다my-env.

```
aws elasticbeanstalk describe-configuration-options --environment-name my-env --
application-name my-app
```

출력(약식):

```
{
    "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"
        }
    ]
}
```

사용 가능한 구성 옵션은 플랫폼 및 구성 버전에 따라 다릅니다. 네임스페이스 및 지원되는 옵션에 대한 자세한 내용은 AWS Elastic Beanstalk 개발자 안내서의 옵션 값을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeConfigurationOptions의 섹션을 참조하세요. AWS CLI

### describe-configuration-settings

다음 코드 예시에서는 describe-configuration-settings을 사용하는 방법을 보여 줍니다.

AWS CLI

환경에 대한 구성 설정을 보려면

다음 명령은 라는 환경에 대한 구성 설정을 검색합니다my-env.

```
aws elasticbeanstalk describe-configuration-settings --environment-name my-env --
application-name my-app
```

출력(약식):

```
{
    "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"
        }
    ]
}
```

네임스페이스 및 지원되는 옵션에 대한 자세한 내용은 AWS Elastic Beanstalk 개발자 안내서의 옵션 값을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeConfigurationSettings의 섹션을 참조하세요. AWS CLI

### `describe-environment-health`

다음 코드 예시에서는 `describe-environment-health`을 사용하는 방법을 보여 줍니다.

AWS CLI

환경 상태를 보려면

다음 명령은 라는 환경에 대한 전체 상태 정보를 검색합니다`my-env`.

```
aws elasticbeanstalk describe-environment-health --environment-name my-env --
attribute-names All
```

출력:

```
{
    "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": []
}
```

상태 정보는 향상된 상태 보고가 활성화된 환경에서만 사용할 수 있습니다. 자세한 내용은 AWS Elastic Beanstalk 개발자 안내서의 향상된 상태 보고 및 모니터링을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeEnvironmentHealth의 섹션을 참조하세요. AWS CLI

### describe-environment-resources

다음 코드 예시에서는 describe-environment-resources을 사용하는 방법을 보여 줍니다.

AWS CLI

환경의 AWS 리소스에 대한 정보를 보려면

다음 명령은 라는 환경의 리소스에 대한 정보를 검색합니다my-env.

```
aws elasticbeanstalk describe-environment-resources --environment-name my-env
```

출력:

```
{
    "EnvironmentResources": {
        "EnvironmentName": "my-env",
        "AutoScalingGroups": [
            {
                "Name": "awseb-e-qu3fyyjyjs-stack-AWSEBAutoScalingGroup-
QSB2ZO88SXZT"
            }
        ],
        "Triggers": [],
        "LoadBalancers": [
            {
                "Name": "awseb-e-q-AWSEBLoa-1EEPZ0K98BIF0"
            }
        ],
        "Queues": [],
        "Instances": [
            {
                "Id": "i-0c91c786"
            }
        ],
        "LaunchConfigurations": [
            {
                "Name": "awseb-e-qu3fyyjyjs-stack-
AWSEBAutoScalingLaunchConfiguration-1UUVQIBC96TQ2"
            }
```

```
        ]
    }
}
```

- 자세한 API 내용은 명령 참조DescribeEnvironmentResources의 섹션을 참조하세요. AWS CLI

### `describe-environments`

다음 코드 예시에서는 `describe-environments`을 사용하는 방법을 보여 줍니다.

AWS CLI

환경에 대한 정보를 보려면

다음 명령은 라는 환경에 대한 정보를 검색합니다`my-env`.

```
aws elasticbeanstalk describe-environments --environment-names my-env
```

출력:

```
{
    "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"
```

```
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeEnvironments의 섹션을 참조하세요. AWS CLI

### `describe-events`

다음 코드 예시에서는 `describe-events`을 사용하는 방법을 보여 줍니다.

AWS CLI

환경에 대한 이벤트를 보려면

다음 명령은 이라는 환경에 대한 이벤트를 검색합니다`my-env`.

```
aws elasticbeanstalk describe-events --environment-name my-env
```

출력(약식):

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeEvents의 섹션을 참조하세요. AWS CLI

### `describe-instances-health`

다음 코드 예시에서는 `describe-instances-health`을 사용하는 방법을 보여 줍니다.

AWS CLI

환경 상태를 보려면

다음 명령은 라는 환경의 인스턴스에 대한 상태 정보를 검색합니다`my-env`.

```
aws elasticbeanstalk describe-instances-health --environment-name my-env --
attribute-names All
```

출력:

```
{
    "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"
}
```


상태 정보는 향상된 상태 보고가 활성화된 환경에서만 사용할 수 있습니다. 자세한 내용은 AWS Elastic Beanstalk 개발자 안내서의 향상된 상태 보고 및 모니터링을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeInstancesHealth의 섹션을 참조하세요. AWS CLI

### list-available-solution-stacks

다음 코드 예시에서는 `list-available-solution-stacks`을 사용하는 방법을 보여 줍니다.

AWS CLI

솔루션 스택을 보려면

다음 명령에는 현재 사용 가능한 모든 플랫폼 구성과 과거에 사용한 모든 에 대한 솔루션 스택이 나열되어 있습니다.

```
aws elasticbeanstalk list-available-solution-stacks
```

출력(약식):

```
{
    "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"
        },
        ...
    ]
}
```

- 자세한 API 내용은 명령 참조ListAvailableSolutionStacks의 섹션을 참조하세요. AWS CLI

### rebuild-environment

다음 코드 예시에서는 rebuild-environment을 사용하는 방법을 보여 줍니다.

AWS CLI

환경을 다시 빌드하려면

다음 명령은 라는 환경에서 리소스를 종료하고 다시 생성합니다my-env.

```
aws elasticbeanstalk rebuild-environment --environment-name my-env
```

- 자세한 API 내용은 명령 참조RebuildEnvironment의 섹션을 참조하세요. AWS CLI

### request-environment-info

다음 코드 예시에서는 request-environment-info을 사용하는 방법을 보여 줍니다.

AWS CLI

테일 로그를 요청하려면

다음 명령은 이라는 환경에서 로그를 요청합니다my-env.

```
aws elasticbeanstalk request-environment-info --environment-name my-env --info-
type tail
```

로그를 요청한 후 를 사용하여 해당 위치를 검색합니다 retrieve-environment-info.

- 자세한 API 내용은 명령 참조RequestEnvironmentInfo의 섹션을 참조하세요. AWS CLI

### `restart-app-server`

다음 코드 예시에서는 `restart-app-server`을 사용하는 방법을 보여 줍니다.

AWS CLI

애플리케이션 서버를 다시 시작하려면

다음 명령은 라는 환경의 모든 인스턴스에서 애플리케이션 서버를 다시 시작합니다`my-env`.

```
aws elasticbeanstalk restart-app-server --environment-name my-env
```

- 자세한 API 내용은 명령 참조RestartAppServer의 섹션을 참조하세요. AWS CLI

### `retrieve-environment-info`

다음 코드 예시에서는 `retrieve-environment-info`을 사용하는 방법을 보여 줍니다.

AWS CLI

테일 로그를 검색하려면

다음 명령은 라는 환경에서 로그에 대한 링크를 검색합니다`my-env`.

```
aws elasticbeanstalk retrieve-environment-info --environment-name my-env --info-
type tail
```

출력:

```
{
    "EnvironmentInfo": [
        {
            "SampleTimestamp": "2015-08-20T22:23:17.703Z",
```

```
            "Message": "https://elasticbeanstalk-us-
west-2-0123456789012.s3.amazonaws.com/resources/environments/
logs/tail/e-fyqyju3yjs/i-09c1c867/TailLogs-1440109397703.out?
AWSAccessKeyId=AKGPT4J56IAJ2EUBL5CQ&Expires=1440195891&Signature=n
%2BEalOV6A2HIOx4Rcfb7LT16bBM%3D",
            "InfoType": "tail",
            "Ec2InstanceId": "i-09c1c867"
        }
    ]
}
```

브라우저에서 링크를 봅니다. 검색하기 전에 를 사용하여 로그를 요청해야 합니다 request-environment-info.

- 자세한 API 내용은 명령 참조RetrieveEnvironmentInfo의 섹션을 참조하세요. AWS CLI

### `swap-environment-cnames`

다음 코드 예시에서는 `swap-environment-cnames`을 사용하는 방법을 보여 줍니다.

AWS CLI

환경을 바꾸려면 CNAMES

다음 명령은 두 환경의 할당된 하위 도메인을 교체합니다.

```
aws elasticbeanstalk swap-environment-cnames --source-environment-name my-env-blue
 --destination-environment-name my-env-green
```

- 자세한 API 내용은 명령 참조SwapEnvironmentCnames의 섹션을 참조하세요. AWS CLI

### `terminate-environment`

다음 코드 예시에서는 `terminate-environment`을 사용하는 방법을 보여 줍니다.

AWS CLI

환경을 종료하려면

다음 명령은 라는 Elastic Beanstalk 환경을 종료합니다`my-env`.

```
aws elasticbeanstalk terminate-environment --environment-name my-env
```

출력:

```
{
    "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"
}
```

- 자세한 API 내용은 명령 참조TerminateEnvironment의 섹션을 참조하세요. AWS CLI

### update-application-version

다음 코드 예시에서는 update-application-version을 사용하는 방법을 보여 줍니다.

AWS CLI

애플리케이션 버전의 설명을 변경하려면

다음 명령은 라는 애플리케이션 버전에 대한 설명을 업데이트합니다22a0-stage-150819_185942.

```
aws elasticbeanstalk update-application-version --version-label 22a0-
stage-150819_185942 --application-name my-app --description "new description"
```

출력:

```
{
```

```
    "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"
        }
    }
}
```

- 자세한 API 내용은 명령 참조UpdateApplicationVersion의 섹션을 참조하세요. AWS CLI

### update-application

다음 코드 예시에서는 update-application을 사용하는 방법을 보여 줍니다.

AWS CLI

애플리케이션의 설명을 변경하려면

다음 명령은 라는 애플리케이션에 대한 설명을 업데이트합니다my-app.

```
aws elasticbeanstalk update-application --application-name my-app --description "my
 Elastic Beanstalk application"
```

출력:

```
{
    "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"
    }
}
```

- 자세한 API 내용은 명령 참조UpdateApplication의 섹션을 참조하세요. AWS CLI

### update-configuration-template

다음 코드 예시에서는 update-configuration-template을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 템플릿을 업데이트하려면

다음 명령은 라는 저장된 구성 템플릿ConfigDocument에서 구성된 CloudWatch 사용자 지정 상태 지표 구성을 제거합니다my-template.

```
aws elasticbeanstalk update-configuration-template --template-
name my-template --application-name my-app --options-to-
remove Namespace=aws:elasticbeanstalk:healthreporting:system,OptionName=ConfigDocument
```

출력:

```
{
    "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"
}
```

네임스페이스 및 지원되는 옵션에 대한 자세한 내용은 AWS Elastic Beanstalk 개발자 안내서의 옵션 값을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateConfigurationTemplate의 섹션을 참조하세요. AWS CLI

### update-environment

다음 코드 예시에서는 update-environment을 사용하는 방법을 보여 줍니다.

AWS CLI

환경을 새 버전으로 업데이트하려면

다음 명령은 'my-env'라는 환경을 속한 애플리케이션의 'v2' 버전으로 업데이트합니다.

```
aws elasticbeanstalk update-environment --environment-name my-env --version-label v2
```

이 명령을 사용하려면 'my-env' 환경이 이미 존재하고 레이블이 'v2'인 유효한 애플리케이션 버전이 있는 애플리케이션에 속해야 합니다.

출력:

```
{
  "ApplicationName": "my-app",
  "EnvironmentName": "my-env",
  "VersionLabel": "v2",
  "Status": "Updating",
  "EnvironmentId": "e-szqipays4h",
  "EndpointURL": "awseb-e-i-AWSEBLoa-1RDLX6TC9VUAO-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"
}
```

환경 변수를 설정하려면

다음 명령은 'my-envPARAM1' 환경의 '' 변수 값을 'ParamValue'로 설정합니다.

```
aws elasticbeanstalk update-environment --environment-name my-env --option-
settings Namespace=aws:elasticbeanstalk:application:environment,OptionName=PARAM1,Value=Para
```

`option-settings` 파라미터는 변수의 이름과 값 외에도 네임스페이스를 사용합니다. Elastic Beanstalk는 환경 변수 외에도 옵션에 대한 여러 네임스페이스를 지원합니다.

파일에서 옵션 설정을 구성하려면

다음 명령은 파일에서 aws:elb:loadbalancer 네임스페이스의 여러 옵션을 구성합니다.

```
aws elasticbeanstalk update-environment --environment-name my-env --option-
settings file://options.json
```

options.json 는 여러 설정을 정의하는 JSON 객체입니다.

```
[
  {
    "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"
  }
]
```

출력:

```
{
    "ApplicationName": "my-app",
    "EnvironmentName": "my-env",
    "VersionLabel": "7f58-stage-150812_025409",
    "Status": "Updating",
    "EnvironmentId": "e-wtp2rpqsej",
    "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"
}
```

네임스페이스 및 지원되는 옵션에 대한 자세한 내용은 AWS Elastic Beanstalk 개발자 안내서의 옵션 값을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateEnvironment의 섹션을 참조하세요. AWS CLI

### validate-configuration-settings

다음 코드 예시에서는 validate-configuration-settings을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 설정을 검증하려면

다음 명령은 CloudWatch 사용자 지정 지표 구성 문서를 검증합니다.

```
aws elasticbeanstalk validate-configuration-settings --application-name my-app --
environment-name my-env --option-settings file://options.json
```

options.json 는 검증할 구성 설정을 하나 이상 포함하는 JSON 문서입니다.

```
[
    {
        "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}"
    }
]
```

지정한 옵션이 지정된 환경에 유효한 경우 Elastic Beanstalk는 빈 메시지 배열을 반환합니다.

```
{
    "Messages": []
}
```

검증에 실패하면 응답에 오류에 대한 정보가 포함됩니다.

```
{
    "Messages": [
        {
            "OptionName": "ConfigDocumet",
            "Message": "Invalid option specification (Namespace:
 'aws:elasticbeanstalk:healthreporting:system', OptionName: 'ConfigDocumet'):
 Unknown configuration setting.",
            "Namespace": "aws:elasticbeanstalk:healthreporting:system",
            "Severity": "error"
        }
    ]
}
```

네임스페이스 및 지원되는 옵션에 대한 자세한 내용은 AWS Elastic Beanstalk 개발자 안내서의 옵션 값을 참조하세요.

- 자세한 API 내용은 명령 참조ValidateConfigurationSettings의 섹션을 참조하세요. AWS CLI

# Elastic Load Balancing - 를 사용하는 버전 1 예제 AWS CLI

다음 코드 예제에서는 Elastic Load Balancing - 버전 1과 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `add-tags`

다음 코드 예시에서는 `add-tags`을 사용하는 방법을 보여 줍니다.

AWS CLI

로드 밸런서에 태그를 추가하려면

이 예제에서는 지정된 로드 밸런서에 태그를 추가합니다.

명령:

```
aws elb add-tags --load-balancer-name my-load-balancer --
tags "Key=project,Value=lima" "Key=department,Value=digital-media"
```

- 자세한 API 내용은 명령 참조AddTags의 섹션을 참조하세요. AWS CLI

### `apply-security-groups-to-load-balancer`

다음 코드 예시에서는 `apply-security-groups-to-load-balancer`을 사용하는 방법을 보여 줍니다.

AWS CLI

보안 그룹을 의 로드 밸런서에 연결하려면 VPC

이 예제에서는 보안 그룹을 의 지정된 로드 밸런서에 연결합니다VPC.

명령:

```
aws elb apply-security-groups-to-load-balancer --load-balancer-name my-load-balancer
 --security-groups sg-fc448899
```

출력:

```
{
  "SecurityGroups": [
      "sg-fc448899"
  ]
}
```

- 자세한 API 내용은 명령 참조ApplySecurityGroupsToLoadBalancer의 섹션을 참조하세요. AWS CLI

### attach-load-balancer-to-subnets

다음 코드 예시에서는 attach-load-balancer-to-subnets을 사용하는 방법을 보여 줍니다.

AWS CLI

서브넷을 로드 밸런서에 연결하려면

이 예제에서는 지정된 로드 밸런서에 대해 구성된 서브넷 세트에 지정된 서브넷을 추가합니다.

명령:

```
aws elb attach-load-balancer-to-subnets --load-balancer-name my-load-balancer --
subnets subnet-0ecac448
```

출력:

```
{
   "Subnets": [
       "subnet-15aaab61",
```

```
        "subnet-0ecac448"
    ]
}
```

- 자세한 API 내용은 명령 참조AttachLoadBalancerToSubnets의 섹션을 참조하세요. AWS CLI

### configure-health-check

다음 코드 예시에서는 configure-health-check을 사용하는 방법을 보여 줍니다.

AWS CLI

백엔드 EC2 인스턴스의 상태 확인 설정을 지정하려면

이 예제에서는 백엔드 EC2 인스턴스의 상태를 평가하는 데 사용되는 상태 확인 설정을 지정합니다.

명령:

```
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
```

출력:

```
{
    "HealthCheck": {
        "HealthyThreshold": 2,
        "Interval": 30,
        "Target": "HTTP:80/png",
        "Timeout": 3,
        "UnhealthyThreshold": 2
    }
}
```

- 자세한 API 내용은 명령 참조ConfigureHealthCheck의 섹션을 참조하세요. AWS CLI

### create-app-cookie-stickiness-policy

다음 코드 예시에서는 create-app-cookie-stickiness-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

HTTPS 로드 밸런서에 대한 고정 정책을 생성하려면

이 예제에서는 애플리케이션 생성 쿠키의 고정 세션 수명을 따르는 고정 정책을 생성합니다.

명령:

```
aws elb create-app-cookie-stickiness-policy --load-balancer-name my-load-balancer --
policy-name my-app-cookie-policy --cookie-name my-app-cookie
```

- 자세한 API 내용은 명령 참조CreateAppCookieStickinessPolicy의 섹션을 참조하세요. AWS CLI

### create-lb-cookie-stickiness-policy

다음 코드 예시에서는 create-lb-cookie-stickiness-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

HTTPS 로드 밸런서에 대한 기간 기반 고정 정책을 생성하려면

이 예제에서는 지정된 만료 기간으로 제어되는 고정 세션 수명과 함께 고정 정책을 생성합니다.

명령:

```
aws elb create-lb-cookie-stickiness-policy --load-balancer-name my-load-balancer --
policy-name my-duration-cookie-policy --cookie-expiration-period 60
```

- 자세한 API 내용은 명령 참조CreateLbCookieStickinessPolicy의 섹션을 참조하세요. AWS CLI

### create-load-balancer-listeners

다음 코드 예시에서는 create-load-balancer-listeners을 사용하는 방법을 보여 줍니다.

AWS CLI

로드 밸런서에 대한 HTTP 리스너를 생성하려면

이 예제에서는 HTTP 프로토콜을 사용하여 포트 80에서 로드 밸런서에 대한 리스너를 생성합니다.

명령:

```
aws elb create-load-balancer-listeners --load-balancer-name my-load-balancer --
listeners "Protocol=HTTP,LoadBalancerPort=80,InstanceProtocol=HTTP,InstancePort=80"
```

로드 밸런서에 대한 HTTPS 리스너를 생성하려면

이 예제에서는 HTTPS 프로토콜을 사용하여 포트 443에서 로드 밸런서에 대한 리스너를 생성합니다.

명령:

```
aws elb create-load-balancer-listeners --load-balancer-name my-load-balancer --
listeners "Protocol=HTTPS,LoadBalancerPort=443,InstanceProtocol=HTTP,InstancePort=80"
```

- 자세한 API 내용은 명령 참조CreateLoadBalancerListeners의 섹션을 참조하세요. AWS CLI

### `create-load-balancer-policy`

다음 코드 예시에서는 `create-load-balancer-policy`을 사용하는 방법을 보여 줍니다.

AWS CLI

로드 밸런서에서 프록시 프로토콜을 활성화하는 정책을 생성하려면

이 예제에서는 지정된 로드 밸런서에서 프록시 프로토콜을 활성화하는 정책을 생성합니다.

명령:

```
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
```

권장 보안 정책을 사용하여 SSL 협상 정책을 생성하려면

이 예제에서는 권장 보안 정책을 사용하여 지정된 HTTPS 로드 SSL 밸런서에 대한 협상 정책을 생성합니다.

명령:

```
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
```

사용자 지정 보안 정책을 사용하여 SSL 협상 정책을 생성하려면

이 예제에서는 프로토콜과 암호를 활성화하여 사용자 지정 보안 정책을 사용하여 HTTPS 로드 밸런서에 대한 SSL 협상 정책을 생성합니다.

명령:

```
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
```

퍼블릭 키 정책을 생성하려면

이 예제에서는 퍼블릭 키 정책을 생성합니다.

명령:

```
aws elb create-load-balancer-policy --load-balancer-name my-load-balancer --policy-name my-PublicKey-policy --policy-type-name PublicKeyPolicyType --policy-attributes AttributeName=PublicKey,AttributeValue=MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQ
+dS74kj//c6x7ROtusUaeQCTgIUkayttRDWchuqo1pHC1u
+n5xxXnBBe2ejbb2WRsKIQ5rXEeixsjFpFsojpSQKkzhVGI6mJVZBJDVKSHmswnwLBdofLhzvllpovBPTHe
+o4haAWvDBALJU0pkSI1FecPHcs2hwxf14zHoXy1e2k36A64nXW43wtfx5qcVSIxtCEOjnYRg7RPvybaGfQ
+v6Iaxb/+7J5kEvZhTFQId+bSiJImF1FSUT1W1xwzBZPUbcUkkXDj45vC2s3Z8E
+Lk7a3uZhvsQHLZnrfuWjBWGWvZ/MhZYgEXAMPLE
```

백엔드 서버 인증 정책을 생성하려면

이 예제에서는 퍼블릭 키 정책을 사용하여 백엔드 인스턴스에서 인증을 활성화하는 백엔드 서버 인증 정책을 생성합니다.

명령:

```
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
```

- 자세한 API 내용은 명령 참조CreateLoadBalancerPolicy의 섹션을 참조하세요. AWS CLI

### create-load-balancer

다음 코드 예시에서는 create-load-balancer을 사용하는 방법을 보여 줍니다.

AWS CLI

HTTP 로드 밸런서를 생성하려면

이 예제에서는 에서 HTTP 리스너가 있는 로드 밸런서를 생성합니다VPC.

명령:

```
aws elb create-load-balancer --load-balancer-name my-load-balancer --
listeners "Protocol=HTTP,LoadBalancerPort=80,InstanceProtocol=HTTP,InstancePort=80"
 --subnets subnet-15aaab61 --security-groups sg-a61988c3
```

출력:

```
{
    "DNSName": "my-load-balancer-1234567890.us-west-2.elb.amazonaws.com"
}
```

이 예제에서는 EC2-Classic에서 HTTP 리스너를 사용하여 로드 밸런서를 생성합니다.

명령:

```
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
```

출력:

```
{
    "DNSName": "my-load-balancer-123456789.us-west-2.elb.amazonaws.com"
}
```

HTTPS 로드 밸런서를 생성하려면

이 예제에서는 에서 HTTPS 리스너를 사용하여 로드 밸런서를 생성합니다VPC.

명령:

```
aws elb create-load-balancer --load-balancer-name my-load-balancer --
listeners "Protocol=HTTP,LoadBalancerPort=80,InstanceProtocol=HTTP,InstancePort=80" "Protoco
certificate/my-server-cert" --subnets subnet-15aaab61 --security-groups sg-a61988c3
```

출력:

```
{
    "DNSName": "my-load-balancer-1234567890.us-west-2.elb.amazonaws.com"
}
```

이 예제에서는 EC2-Classic에서 HTTPS 리스너를 사용하여 로드 밸런서를 생성합니다.

명령:

```
aws elb create-load-balancer --load-balancer-name my-load-balancer --
listeners "Protocol=HTTP,LoadBalancerPort=80,InstanceProtocol=HTTP,InstancePort=80" "Protoco
certificate/my-server-cert" --availability-zones us-west-2a us-west-2b
```

출력:

```
{
    "DNSName": "my-load-balancer-123456789.us-west-2.elb.amazonaws.com"
}
```

내부 로드 밸런서를 생성하려면

이 예제에서는 에서 HTTP 리스너가 있는 내부 로드 밸런서를 생성합니다VPC.

명령:

```
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
```

출력:

```
{
    "DNSName": "internal-my-load-balancer-123456789.us-west-2.elb.amazonaws.com"
}
```

- 자세한 API 내용은 명령 참조CreateLoadBalancer의 섹션을 참조하세요. AWS CLI

### delete-load-balancer-listeners

다음 코드 예시에서는 delete-load-balancer-listeners을 사용하는 방법을 보여 줍니다.

AWS CLI

로드 밸런서에서 리스너를 삭제하려면

이 예제에서는 지정된 로드 밸런서에서 지정된 포트의 리스너를 삭제합니다.

명령:

```
aws elb delete-load-balancer-listeners --load-balancer-name my-load-balancer --load-balancer-ports 80
```

- 자세한 API 내용은 명령 참조DeleteLoadBalancerListeners의 섹션을 참조하세요. AWS CLI

### delete-load-balancer-policy

다음 코드 예시에서는 delete-load-balancer-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

로드 밸런서에서 정책을 삭제하려면

이 예제에서는 지정된 로드 밸런서에서 지정된 정책을 삭제합니다. 리스너에서 정책을 활성화해서는 안 됩니다.

명령:

```
aws elb delete-load-balancer-policy --load-balancer-name my-load-balancer --policy-name my-duration-cookie-policy
```

- 자세한 API 내용은 명령 참조DeleteLoadBalancerPolicy의 섹션을 참조하세요. AWS CLI

### delete-load-balancer

다음 코드 예시에서는 delete-load-balancer을 사용하는 방법을 보여 줍니다.

AWS CLI

로드 밸런서를 삭제하는 방법

이 예제에서는 지정된 로드 밸런서를 삭제합니다.

명령:

```
aws elb delete-load-balancer --load-balancer-name my-load-balancer
```

- 자세한 API 내용은 명령 참조DeleteLoadBalancer의 섹션을 참조하세요. AWS CLI

### deregister-instances-from-load-balancer

다음 코드 예시에서는 deregister-instances-from-load-balancer을 사용하는 방법을 보여 줍니다.

AWS CLI

로드 밸런서에서 인스턴스 등록을 취소하려면

이 예제에서는 지정된 로드 밸런서에서 지정된 인스턴스의 등록을 취소합니다.

명령:

```
aws elb deregister-instances-from-load-balancer --load-balancer-name my-load-
balancer --instances i-d6f6fae3
```

출력:

```
{
    "Instances": [
        {
            "InstanceId": "i-207d9717"
        },
        {
            "InstanceId": "i-afefb49b"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DeregisterInstancesFromLoadBalancer의 섹션을 참조하세요. AWS CLI

### `describe-account-limits`

다음 코드 예시에서는 `describe-account-limits`을 사용하는 방법을 보여 줍니다.

AWS CLI

Classic Load Balancer 제한 설명

다음 `describe-account-limits` 예제에서는 AWS 계정의 Classic Load Balancer 제한에 대한 세부 정보를 보여줍니다.

```
aws elb describe-account-limits
```

출력:

```
{
    "Limits": [
        {
            "Name": "classic-load-balancers",
            "Max": "20"
        },
        {
            "Name": "classic-listeners",
            "Max": "100"
        },
        {
            "Name": "classic-registered-instances",
            "Max": "1000"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeAccountLimits의 섹션을 참조하세요. AWS CLI

### `describe-instance-health`

다음 코드 예시에서는 `describe-instance-health`을 사용하는 방법을 보여 줍니다.

AWS CLI

로드 밸런서 인스턴스의 상태를 설명하려면

이 예제에서는 지정된 로드 밸런서에 대한 인스턴스의 상태를 설명합니다.

명령:

```
aws elb describe-instance-health --load-balancer-name my-load-balancer
```

출력:

```
{
  "InstanceStates": [
      {
          "InstanceId": "i-207d9717",
          "ReasonCode": "N/A",
          "State": "InService",
          "Description": "N/A"
      },
      {
          "InstanceId": "i-afefb49b",
          "ReasonCode": "N/A",
          "State": "InService",
          "Description": "N/A"
      }
  ]
}
```

로드 밸런서 인스턴스의 상태를 설명하려면

이 예제에서는 지정된 로드 밸런서에 대해 지정된 인스턴스의 상태를 설명합니다.

명령:

```
aws elb describe-instance-health --load-balancer-name my-load-balancer --
instances i-7299c809
```

다음은 등록 중인 인스턴스에 대한 예제 응답입니다.

출력:

```
{
```

```
  "InstanceStates": [
      {
          "InstanceId": "i-7299c809",
          "ReasonCode": "ELB",
          "State": "OutOfService",
          "Description": "Instance registration is still in progress."
    }
  ]
}
```

다음은 비정상 인스턴스에 대한 예제 응답입니다.

출력:

```
{
  "InstanceStates": [
      {
          "InstanceId": "i-7299c809",
          "ReasonCode": "Instance",
          "State": "OutOfService",
          "Description": "Instance has failed at least the UnhealthyThreshold number
 of health checks consecutively."
      }
  ]
}
```

- 자세한 API 내용은 명령 참조DescribeInstanceHealth의 섹션을 참조하세요. AWS CLI

### `describe-load-balancer-attributes`

다음 코드 예시에서는 `describe-load-balancer-attributes`을 사용하는 방법을 보여 줍니다.

AWS CLI

로드 밸런서의 속성을 설명하려면

이 예제에서는 지정된 로드 밸런서의 속성을 설명합니다.

명령:

```
aws elb describe-load-balancer-attributes --load-balancer-name my-load-balancer
```

출력:

```
{
  "LoadBalancerAttributes": {
      "ConnectionDraining": {
          "Enabled": false,
          "Timeout": 300
      },
      "CrossZoneLoadBalancing": {
          "Enabled": true
      },
      "ConnectionSettings": {
          "IdleTimeout": 30
      },
      "AccessLog": {
          "Enabled": false
    }
  }
}
```

- 자세한 API 내용은 명령 참조DescribeLoadBalancerAttributes의 섹션을 참조하세요. AWS CLI

### `describe-load-balancer-policies`

다음 코드 예시에서는 `describe-load-balancer-policies`을 사용하는 방법을 보여 줍니다.

AWS CLI

로드 밸런서와 연결된 모든 정책을 설명하려면

이 예제에서는 지정된 로드 밸런서와 연결된 모든 정책을 설명합니다.

명령:

```
aws elb describe-load-balancer-policies --load-balancer-name my-load-balancer
```

출력:

```
{
  "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"
      },
      .
      .
      .
  ]
}
```

로드 밸런서와 연결된 특정 정책을 설명하려면

이 예제에서는 지정된 로드 밸런서와 연결된 지정된 정책을 설명합니다.

명령:

```
aws elb describe-load-balancer-policies --load-balancer-name my-load-balancer --
policy-name my-authentication-policy
```

출력:

```
{
  "PolicyDescriptions": [
      {
          "PolicyAttributeDescriptions": [
              {
                  "AttributeName": "PublicKeyPolicyName",
                  "AttributeValue": "my-PublicKey-policy"
              }
          ],
          "PolicyName": "my-authentication-policy",
          "PolicyTypeName": "BackendServerAuthenticationPolicyType"
      }
  ]
}
```

- 자세한 API 내용은 명령 참조DescribeLoadBalancerPolicies의 섹션을 참조하세요. AWS CLI

### `describe-load-balancer-policy-types`

다음 코드 예시에서는 `describe-load-balancer-policy-types`을 사용하는 방법을 보여 줍니다.

AWS CLI

Elastic Load Balancing에서 정의한 로드 밸런서 정책 유형을 설명하려면

이 예제에서는 로드 밸런서에 대한 정책 구성을 생성하는 데 사용할 수 있는 로드 밸런서 정책 유형을 설명합니다.

명령:

```
aws elb describe-load-balancer-policy-types
```

출력:

```
{
  "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."
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeLoadBalancerPolicyTypes의 섹션을 참조하세요. AWS CLI

### `describe-load-balancers`

다음 코드 예시에서는 `describe-load-balancers`을 사용하는 방법을 보여 줍니다.

AWS CLI

로드 밸런서를 설명하려면

이 예시에서는 모든 로드 밸런서를 설명합니다.

명령:

```
aws elb describe-load-balancers
```

로드 밸런서 중 하나를 설명하려면

이 예시에서는 지정된 로드 밸런서를 설명합니다.

명령:

```
aws elb describe-load-balancers --load-balancer-name my-load-balancer
```

다음 예제 응답은 의 HTTPS 로드 밸런서에 대한 것입니다VPC.

출력:

```
{
  "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"
        }
      }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeLoadBalancers의 섹션을 참조하세요. AWS CLI

### describe-tags

다음 코드 예시에서는 describe-tags을 사용하는 방법을 보여 줍니다.

AWS CLI

로드 밸런서에 할당된 태그를 설명하려면

이 예제에서는 지정된 로드 밸런서에 할당된 태그를 설명합니다.

명령:

```
aws elb describe-tags --load-balancer-name my-load-balancer
```

출력:

```
{
  "TagDescriptions": [
      {
```

```
            "Tags": [
                {
                    "Value": "lima",
                    "Key": "project"
                },
                {
                    "Value": "digital-media",
                    "Key": "department"
                }
            ],
            "LoadBalancerName": "my-load-balancer"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeTags의 섹션을 참조하세요. AWS CLI

### `detach-load-balancer-from-subnets`

다음 코드 예시에서는 `detach-load-balancer-from-subnets`을 사용하는 방법을 보여 줍니다.

AWS CLI

서브넷에서 로드 밸런서를 분리하려면

이 예제에서는 지정된 서브넷에서 지정된 로드 밸런서를 분리합니다.

명령:

```
aws elb detach-load-balancer-from-subnets --load-balancer-name my-load-balancer --
subnets subnet-0ecac448
```

출력:

```
{
   "Subnets": [
       "subnet-15aaab61"
   ]
}
```

- 자세한 API 내용은 명령 참조DetachLoadBalancerFromSubnets의 섹션을 참조하세요. AWS CLI

### disable-availability-zones-for-load-balancer

다음 코드 예시에서는 disable-availability-zones-for-load-balancer을 사용하는 방법을 보여 줍니다.

AWS CLI

로드 밸런서의 가용 영역을 비활성화하려면

이 예제에서는 지정된 로드 밸런서에 대한 가용 영역 세트에서 지정된 가용 영역을 제거합니다.

명령:

```
aws elb disable-availability-zones-for-load-balancer --load-balancer-name my-load-balancer --availability-zones us-west-2a
```

출력:

```
{
    "AvailabilityZones": [
        "us-west-2b"
    ]
}
```

- 자세한 API 내용은 명령 참조DisableAvailabilityZonesForLoadBalancer의 섹션을 참조하세요. AWS CLI

### enable-availability-zones-for-load-balancer

다음 코드 예시에서는 enable-availability-zones-for-load-balancer을 사용하는 방법을 보여 줍니다.

AWS CLI

로드 밸런서에 대해 가용 영역을 활성화하려면

이 예제에서는 지정된 가용 영역을 지정된 로드 밸런서에 추가합니다.

명령:

```
aws elb enable-availability-zones-for-load-balancer --load-balancer-name my-load-balancer --availability-zones us-west-2b
```

출력:

```
{
    "AvailabilityZones": [
        "us-west-2a",
        "us-west-2b"
    ]
}
```

- 자세한 API 내용은 명령 참조EnableAvailabilityZonesForLoadBalancer의 섹션을 참조하세요. AWS CLI

### `modify-load-balancer-attributes`

다음 코드 예시에서는 `modify-load-balancer-attributes`을 사용하는 방법을 보여 줍니다.

AWS CLI

로드 밸런서의 속성을 수정하려면

이 예제에서는 지정된 로드 밸런서의 `CrossZoneLoadBalancing` 속성을 수정합니다.

명령:

```
aws elb modify-load-balancer-attributes --load-balancer-name my-load-balancer --
load-balancer-attributes "{\"CrossZoneLoadBalancing\":{\"Enabled\":true}}"
```

출력:

```
{
    "LoadBalancerAttributes": {
        "CrossZoneLoadBalancing": {
            "Enabled": true
        }
    },
    "LoadBalancerName": "my-load-balancer"
}
```

이 예제에서는 지정된 로드 밸런서의 `ConnectionDraining` 속성을 수정합니다.

명령:

```
aws elb modify-load-balancer-attributes --load-balancer-name my-load-balancer
 --load-balancer-attributes "{\"ConnectionDraining\":{\"Enabled\":true,\"Timeout
\":300}}"
```

출력:

```
{
    "LoadBalancerAttributes": {
        "ConnectionDraining": {
            "Enabled": true,
            "Timeout": 300
        }
    },
    "LoadBalancerName": "my-load-balancer"
}
```

- 자세한 API 내용은 명령 참조ModifyLoadBalancerAttributes의 섹션을 참조하세요. AWS CLI

### register-instances-with-load-balancer

다음 코드 예시에서는 register-instances-with-load-balancer을 사용하는 방법을 보여 줍니다.

AWS CLI

로드 밸런서에 인스턴스를 등록하려면

이 예제에서는 지정된 인스턴스를 지정된 로드 밸런서에 등록합니다.

명령:

```
aws elb register-instances-with-load-balancer --load-balancer-name my-load-balancer
 --instances i-d6f6fae3
```

출력:

```
{
   "Instances": [
       {
           "InstanceId": "i-d6f6fae3"
```

```
        },
        {
            "InstanceId": "i-207d9717"
        },
        {
            "InstanceId": "i-afefb49b"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조RegisterInstancesWithLoadBalancer의 섹션을 참조하세요. AWS CLI

### `remove-tags`

다음 코드 예시에서는 `remove-tags`을 사용하는 방법을 보여 줍니다.

AWS CLI

로드 밸런서에서 태그를 제거하려면

이 예제에서는 지정된 로드 밸런서에서 태그를 제거합니다.

명령:

```
aws elb remove-tags --load-balancer-name my-load-balancer --tags project
```

- 자세한 API 내용은 명령 참조RemoveTags의 섹션을 참조하세요. AWS CLI

### `set-load-balancer-listener-ssl-certificate`

다음 코드 예시에서는 `set-load-balancer-listener-ssl-certificate`을 사용하는 방법을 보여 줍니다.

AWS CLI

HTTPS 로드 밸런서의 SSL 인증서를 업데이트하려면

이 예제는 지정된 HTTPS 로드 밸런서에 대한 기존 SSL 인증서를 대체합니다.

명령:

```
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
```

- 자세한 API 내용은 명령 참조SetLoadBalancerListenerSslCertificate의 섹션을 참조하세요. AWS CLI

### set-load-balancer-policies-for-backend-server

다음 코드 예시에서는 set-load-balancer-policies-for-backend-server을 사용하는 방법을 보여 줍니다.

AWS CLI

백엔드 인스턴스의 포트와 연결된 정책을 교체하려면

이 예제는 현재 지정된 포트와 연결된 정책을 대체합니다.

명령:

```
aws elb set-load-balancer-policies-for-backend-server --load-balancer-name my-load-
balancer --instance-port 80 --policy-names my-ProxyProtocol-policy
```

백엔드 인스턴스의 포트와 현재 연결된 모든 정책을 제거하려면

이 예제에서는 지정된 포트와 연결된 모든 정책을 제거합니다.

명령:

```
aws elb set-load-balancer-policies-for-backend-server --load-balancer-name my-load-
balancer --instance-port 80 --policy-names []
```

정책이 제거되었는지 확인하려면 describe-load-balancer-policies 명령을 사용합니다.

- 자세한 API 내용은 명령 참조SetLoadBalancerPoliciesForBackendServer의 섹션을 참조하세요. AWS CLI

### set-load-balancer-policies-of-listener

다음 코드 예시에서는 set-load-balancer-policies-of-listener을 사용하는 방법을 보여 줍니다.

AWS CLI

리스너와 연결된 정책을 바꾸려면

이 예제는 현재 지정된 리스너와 연결된 정책을 대체합니다.

명령:

```
aws elb set-load-balancer-policies-of-listener --load-balancer-name my-load-balancer
 --load-balancer-port 443 --policy-names my-SSLNegotiation-policy
```

리스너와 연결된 모든 정책을 제거하려면

이 예제에서는 현재 지정된 리스너와 연결된 모든 정책을 제거합니다.

명령:

```
aws elb set-load-balancer-policies-of-listener --load-balancer-name my-load-balancer
 --load-balancer-port 443 --policy-names []
```

로드 밸런서에서 정책이 제거되었는지 확인하려면 `describe-load-balancer-policies` 명령을 사용합니다.

- 자세한 API 내용은 명령 참조SetLoadBalancerPoliciesOfListener의 섹션을 참조하세요. AWS CLI

## Elastic Load Balancing - 를 사용한 버전 2 예제 AWS CLI

다음 코드 예제에서는 Elastic Load Balancing - 버전 2와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### `add-listener-certificates`

다음 코드 예시에서는 `add-listener-certificates`을 사용하는 방법을 보여 줍니다.

AWS CLI

보안 리스너에 인증서를 추가하려면

이 예제에서는 지정된 보안 리스너에 지정된 인증서를 추가합니다.

명령:

```
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
```

출력:

```
{
  "Certificates": [
      {
          "CertificateArn": "arn:aws:acm:us-west-2:123456789012:certificate/5cc54884-f4a3-4072-80be-05b9ba72f705",
          "IsDefault": false
      }
  ]
}
```

- 자세한 API 내용은 명령 참조AddListenerCertificates의 섹션을 참조하세요. AWS CLI

### `add-tags`

다음 코드 예시에서는 `add-tags`을 사용하는 방법을 보여 줍니다.

AWS CLI

로드 밸런서에 태그를 추가하려면

다음 `add-tags` 예제에서는 지정된 로드 밸런서에 `project` 및 `department` 태그를 추가합니다.

```
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"
```

- 자세한 API 내용은 명령 참조AddTags의 섹션을 참조하세요. AWS CLI

### `create-listener`

다음 코드 예시에서는 `create-listener`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: HTTP 리스너 생성

다음 `create-listener` 예제에서는 지정된 대상 그룹에 요청을 전달하는 지정된 Application Load Balancer에 대한 HTTP리스너를 생성합니다.

```
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
```

자세한 내용은 Application Load Balancer 사용 설명서의 자습서: 를 사용하여 Application Load Balancer 생성을 AWS CLI 참조하세요.

예제 2: HTTPS 리스너 생성

다음 `create-listener` 예제에서는 지정된 대상 그룹에 요청을 전달하는 지정된 Application Load Balancer에 대한 HTTPS리스너를 생성합니다. HTTPS 리스너에 대한 SSL 인증서를 지정해야 합니다. AWS Certificate Manager()를 사용하여 인증서를 생성하고 관리할 수 있습니다ACM. 또는 SSL/TLS 도구를 사용하여 인증서를 생성하고, 인증 기관(CA)에서 서명한 인증서를 가져오고, 인증서를 AWS Identity and Access Management()에 업로드할 수 있습니다IAM.

```
aws elbv2 create-listener \
    --load-balancer-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 \
```

```
    --protocol HTTPS \
    --port 443 \
    --certificates CertificateArn=arn:aws:acm:us-
west-2:123456789012:certificate/3dcb0a41-bd72-4774-9ad9-756919c40557 \
    --ssl-policy ELBSecurityPolicy-2016-08 \
    --default-actions Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

자세한 내용은 Application Load Balancer 사용 설명서의 HTTPS 리스너 추가를 참조하세요.

예제 3: TCP 리스너 생성

다음 `create-listener` 예제에서는 요청을 지정된 대상 그룹에 전달하는 지정된 Network Load Balancer에 대한 TCP리스너를 생성합니다.

```
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
```

자세한 내용은 Network Load Balancer 사용 설명서의 자습서: 를 사용하여 Network Load Balancer 생성을 AWS CLI 참조하세요.

예제 4: TLS 리스너 생성

다음 `create-listener` 예제에서는 지정된 대상 그룹에 요청을 전달하는 지정된 Network Load Balancer에 대한 TLS리스너를 생성합니다. TLS 리스너에 대한 SSL 인증서를 지정해야 합니다.

```
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
```

자세한 내용은 TLS Network Load Balancer 사용 설명서의 Network Load Balancer 리스너를 참조하세요.

예제 5: UDP 리스너 생성

다음 `create-listener` 예제에서는 지정된 대상 그룹에 요청을 전달하는 지정된 Network Load Balancer에 대한 UDP리스너를 생성합니다.

```
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
```

자세한 내용은 Network Load Balancer 사용 설명서의 자습서: 를 사용하여 AWS CLI Network Load Balancer 생성을 참조하세요.

예 6: 지정된 게이트웨이 및 전달을 위한 리스너를 생성하는 방법

다음 `create-listener` 예시에서는 요청을 지정된 대상 그룹으로 전달하는 지정된 Gateway Load Balancer의 리스너를 생성합니다.

```
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
```

출력:

```
{
    "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"
                            }
                        ]
                    }
                }
            ]
        }
    ]
}
```

자세한 내용은 Gateway Load Balancer 사용 설명서의 를 사용하여 AWS CLI Gateway Load Balancer 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조CreateListener의 섹션을 참조하세요. AWS CLI

### `create-load-balancer`

다음 코드 예시에서는 `create-load-balancer`을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: 인터넷 경계 로드 밸런서를 생성하는 방법

다음 `create-load-balancer` 예시에서는 인터넷 경계 Application Load Balancer를 생성하고 지정된 서브넷의 가용 영역을 활성화합니다.

```
aws elbv2 create-load-balancer \
    --name my-load-balancer \
    --subnets subnet-b7d581c0 subnet-8360a9e7
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Application Load Balancer 사용 설명서의 자습서: 를 사용하여 Application Load Balancer 생성을 AWS CLI 참조하세요.

예 2: 내부 로드 밸런서를 생성하는 방법

다음 `create-load-balancer` 예시에서는 내부 Application Load Balancer를 생성하고 지정된 서브넷의 가용 영역을 활성화합니다.

```
aws elbv2 create-load-balancer \
    --name my-internal-load-balancer \
    --scheme internal \
    --subnets subnet-b7d581c0 subnet-8360a9e7
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Application Load Balancer 사용 설명서의 자습서: 를 사용하여 AWS CLI Application Load Balancer 생성을 참조하세요.

예 3: Network Load Balancer를 생성하는 방법

다음 `create-load-balancer` 예시에서는 인터넷 경계 Network Load Balancer를 생성하고 지정된 서브넷의 가용 영역을 활성화합니다. 서브넷 매핑을 사용하여 지정된 탄력적 IP 주소를 로드 밸런서 노드가 가용 영역에 사용하는 네트워크 인터페이스와 연결합니다.

```
aws elbv2 create-load-balancer \
    --name my-network-load-balancer \
    --type network \
    --subnet-mappings SubnetId=subnet-b7d581c0,AllocationId=eipalloc-64d5890a
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Network Load Balancer 사용 설명서의 자습서: 를 사용하여 AWS CLI Network Load Balancer 생성을 참조하세요.

예 4: Gateway Load Balancer를 생성하는 방법

다음 `create-load-balancer` 예시에서는 Gateway Load Balancer를 생성하고 지정된 서브넷의 가용 영역을 활성화합니다.

```
aws elbv2 create-load-balancer \
    --name my-gateway-load-balancer \
    --type gateway \
    --subnets subnet-dc83f691 subnet-a62583f9
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Gateway Load Balancer 사용 설명서의 를 사용하여 AWS CLI Gateway Load Balancer 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조CreateLoadBalancer의 섹션을 참조하세요. AWS CLI

### create-rule

다음 코드 예시에서는 create-rule을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 경로 조건 및 전달 작업을 사용하여 규칙을 생성하려면

다음 create-rule 예제에서는 에 지정된 패턴이 URL 포함된 경우 지정된 대상 그룹에 요청을 전달하는 규칙을 생성합니다.

```
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
```

conditions-pattern.json의 콘텐츠:

```
[
    {
        "Field": "path-pattern",
        "PathPatternConfig": {
            "Values": ["/images/*"]
        }
    }
]
```

예제 2: 호스트 조건과 고정 응답을 사용하여 규칙을 생성하려면

다음 create-rule 예제에서는 호스트 헤더의 호스트 이름이 지정된 호스트 이름과 일치하는 경우 고정 응답을 제공하는 규칙을 생성합니다.

```
aws elbv2 create-rule \
    --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/
my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2 \
    --priority 10 \
    --conditions file://conditions-host.json \
    --actions file://actions-fixed-response.json
```

conditions-host.json의 콘텐츠

```
[
  {
      "Field": "host-header",
      "HostHeaderConfig": {
          "Values": ["*.example.com"]
      }
  }
]
```

actions-fixed-response.json의 콘텐츠

```
[
    {
        "Type": "fixed-response",
        "FixedResponseConfig": {
            "MessageBody": "Hello world",
            "StatusCode": "200",
            "ContentType": "text/plain"
        }
    }
]
```

예제 3: 소스 IP 주소 조건, 인증 작업 및 전달 작업을 사용하여 규칙을 생성하려면

다음 create-rule 예제에서는 소스 IP 주소가 지정된 IP 주소와 일치하는 경우 사용자를 인증하는 규칙을 생성하고 인증이 성공하면 지정된 대상 그룹에 요청을 전달합니다.

```
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
```

conditions-source-ip.json의 콘텐츠

```
[
    {
        "Field": "source-ip",
```

```
        "SourceIpConfig": {
            "Values": ["192.0.2.0/24", "198.51.100.10/32"]
        }
    }
]
```

actions-authenticate.json의 콘텐츠

```
[
    {
        "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": "abcdefghijklmnopqrstuvwxyz123456789",
            "ClientSecret": "12345678901234567890123456789O",
            "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
    }
]
```

- 자세한 API 내용은 명령 참조CreateRule의 섹션을 참조하세요. AWS CLI

### create-target-group

다음 코드 예시에서는 create-target-group을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: Application Load Balancer의 대상 그룹을 생성하려면

다음 `create-target-group` 예시에서는 인스턴스 ID(대상 유형 `instance`)별로 대상을 등록하는 Application Load Balancer의 대상 그룹을 생성합니다. 이 대상 그룹은 HTTP 프로토콜, 포트 80 및 HTTP 대상 그룹의 기본 상태 확인 설정을 사용합니다.

```
aws elbv2 create-target-group \
    --name my-targets \
    --protocol HTTP \
    --port 80 \
    --target-type instance \
    --vpc-id vpc-3ac0fb5f
```

출력:

```
{
    "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"
        }
    ]
```

```
}
```

자세한 내용은 Application Load Balancer 사용 설명서의 대상 그룹 생성을 참조하세요.

예제 2: Application Load Balancer에서 Lambda 함수로 트래픽을 라우팅하는 대상 그룹을 생성하려면

다음 `create-target-group` 예시에서는 대상이 Lambda 함수(대상 유형 `lambda`)인 Application Load Balancer의 대상 그룹을 생성합니다. 기본적으로 상태 확인은 이 대상 그룹에 대해 비활성화됩니다.

```
aws elbv2 create-target-group \
    --name my-lambda-target \
    --target-type lambda
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Application Load Balancer 사용 설명서의 Lambda 함수를 대상으로를 참조하세요.

예제 3: Network Load Balancer의 대상 그룹을 생성하려면

다음 `create-target-group` 예시에서는 IP 주소(대상 유형 `ip`)별로 대상을 등록하는 Network Load Balancer의 대상 그룹을 생성합니다. 이 대상 그룹은 TCP 프로토콜, 포트 80 및 TCP 대상 그룹의 기본 상태 확인 설정을 사용합니다.

```
aws elbv2 create-target-group \
    --name my-ip-targets \
    --protocol TCP \
    --port 80 \
    --target-type ip \
    --vpc-id vpc-3ac0fb5f
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Network Load Balancer 사용 설명서의 대상 그룹 생성을 참조하세요.

예제 4: Network Load Balancer에서 Application Load Balancer로 트래픽을 라우팅하는 대상 그룹을 생성하려면

다음 `create-target-group` 예제에서는 Application Load Balancer를 대상으로 등록하는 Network Load Balancer의 대상 그룹을 생성합니다(대상 유형은 임`alb`). Application Load Balancer

aws elbv2 create-target-group --name my-alb-target --protocol TCP --port 80 --target-type alb --vpc-id vpc-3ac0fb5f

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Network Load Balancer 사용 설명서의 Application Load Balancer를 대상으로 하는 대상 그룹 생성을 참조하세요.

예제 5: Gateway Load Balancer의 대상 그룹을 생성하려면

다음 `create-target-group` 예제에서는 대상이 인스턴스이고 대상 그룹 프로토콜이 인 Gateway Load Balancer의 대상 그룹을 생성합니다`GENEVE`.

```
aws elbv2 create-target-group \
    --name my-glb-targetgroup \
    --protocol GENEVE \
    --port 6081 \
```

```
    --target-type instance \
    --vpc-id vpc-838475fe
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Gateway Load Balancer 사용 설명서의 대상 그룹 <https://docs.aws.amazon.com/elasticloadbalancing/latest/gateway/create-target-group.html>`__ 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateTargetGroup의 섹션을 참조하세요. AWS CLI

### `delete-listener`

다음 코드 예시에서는 `delete-listener`을 사용하는 방법을 보여 줍니다.

AWS CLI

리스너를 삭제하려면

다음 `delete-listener` 예제에서는 지정된 리스너를 삭제합니다.

```
aws elbv2 delete-listener \
```

```
    --listener-arn arn:aws:elasticloadbalancing:ua-west-2:123456789012:listener/app/
my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2
```

- 자세한 API 내용은 명령 참조DeleteListener의 섹션을 참조하세요. AWS CLI

### delete-load-balancer

다음 코드 예시에서는 delete-load-balancer을 사용하는 방법을 보여 줍니다.

AWS CLI

로드 밸런서를 삭제하는 방법

다음 delete-load-balancer 예시에서는 지정된 로드 밸런서를 삭제합니다.

```
aws elbv2 delete-load-balancer \
    --load-balancer-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188
```

- 자세한 API 내용은 명령 참조DeleteLoadBalancer의 섹션을 참조하세요. AWS CLI

### delete-rule

다음 코드 예시에서는 delete-rule을 사용하는 방법을 보여 줍니다.

AWS CLI

규칙을 삭제하려면

다음 delete-rule 예제에서는 지정된 규칙을 삭제합니다.

```
aws elbv2 delete-rule \
    --rule-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener-rule/
app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2/1291d13826f405c3
```

- 자세한 API 내용은 명령 참조DeleteRule의 섹션을 참조하세요. AWS CLI

### delete-target-group

다음 코드 예시에서는 delete-target-group을 사용하는 방법을 보여 줍니다.

AWS CLI

대상 그룹을 삭제하는 방법

다음 delete-target-group 예시에서는 지정된 대상 그룹을 삭제합니다.

```
aws elbv2 delete-target-group \
    --target-group-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Application Load Balancer Load Balancer 안내서의 로드 밸런서 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteTargetGroup의 섹션을 참조하세요. AWS CLI

### deregister-targets

다음 코드 예시에서는 deregister-targets을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 대상 그룹에서 대상 등록을 취소하려면

다음 deregister-targets 예제에서는 지정된 대상 그룹에서 지정된 인스턴스를 제거합니다.

```
aws elbv2 deregister-targets \
    --target-group-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067 \
    --targets Id=i-1234567890abcdef0
```

예제 2: 포트 재정의를 사용하여 등록된 대상 등록 취소

다음 deregister-targets 예제에서는 포트 재정의를 사용하여 등록된 대상 그룹에서 인스턴스를 제거합니다.

```
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
```

- 자세한 API 내용은 명령 참조DeregisterTargets의 섹션을 참조하세요. AWS CLI

### describe-account-limits

다음 코드 예시에서는 describe-account-limits을 사용하는 방법을 보여 줍니다.

AWS CLI

Elastic Load Balancing 제한 설명

다음 describe-account-limits 예제에서는 현재 리전의 AWS 계정에 대한 Elastic Load Balancing 제한을 표시합니다.

```
aws elbv2 describe-account-limits
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS 일반 참조 의 할당량을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeAccountLimits의 섹션을 참조하세요. AWS CLI

### `describe-listener-certificates`

다음 코드 예시에서는 `describe-listener-certificates`을 사용하는 방법을 보여 줍니다.

AWS CLI

보안 리스너의 인증서를 설명하려면

이 예제에서는 지정된 보안 리스너의 인증서를 설명합니다.

명령:

```
aws elbv2 describe-listener-certificates --listener-
arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-load-
balancer/50dc6c495c0c9188/f2f7dc8efc522ab2
```

출력:

```
{
  "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
      }
  ]
}
```

- 자세한 API 내용은 명령 참조DescribeListenerCertificates의 섹션을 참조하세요. AWS CLI

### describe-listeners

다음 코드 예시에서는 describe-listeners을 사용하는 방법을 보여 줍니다.

AWS CLI

리스너를 설명하려면

이 예제에서는 지정된 리스너를 설명합니다.

명령:

```
aws elbv2 describe-listeners --listener-arns arn:aws:elasticloadbalancing:us-
west-2:123456789012:listener/app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2
```

출력:

```
{
  "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"
      }
  ]
}
```

로드 밸런서의 리스너를 설명하려면

이 예제에서는 지정된 로드 밸런서에 대한 리스너를 설명합니다.

명령:

```
aws elbv2 describe-listeners --load-balancer-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188
```

출력:

```
{
  "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeListeners의 섹션을 참조하세요. AWS CLI

### `describe-load-balancer-attributes`

다음 코드 예시에서는 `describe-load-balancer-attributes`을 사용하는 방법을 보여 줍니다.

AWS CLI

로드 밸런서 속성을 설명하려면

다음 `describe-load-balancer-attributes` 예제에서는 지정된 로드 밸런서의 속성을 표시합니다.

```
aws elbv2 describe-load-balancer-attributes \
    --load-balancer-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188
```

다음 예제 출력은 Application Load Balancer 의 속성을 보여줍니다.

```
{
    "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"
        }
    ]
}
```

다음 예제 출력에는 Network Load Balancer에 대한 속성이 포함됩니다.

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeLoadBalancerAttributes의 섹션을 참조하세요. AWS CLI

### `describe-load-balancers`

다음 코드 예시에서는 `describe-load-balancers`을 사용하는 방법을 보여 줍니다.

AWS CLI

로드 밸런서를 설명하는 방법

이 예시에서는 지정된 로드 밸런서를 설명합니다.

명령:

```
aws elbv2 describe-load-balancers --load-balancer-
arns arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/my-load-
balancer/50dc6c495c0c9188
```

출력:

```
{
  "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"
      }
  ]
}
```

모든 로드 밸런서를 설명하는 방법

이 예시에서는 모든 로드 밸런서를 설명합니다.

명령:

```
aws elbv2 describe-load-balancers
```

- 자세한 API 내용은 명령 참조DescribeLoadBalancers의 섹션을 참조하세요. AWS CLI

### describe-rules

다음 코드 예시에서는 describe-rules을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 규칙 설명

다음 describe-rules 예제에서는 지정된 규칙에 대한 세부 정보를 표시합니다.

```
aws elbv2 describe-rules \
    --rule-arns arn:aws:elasticloadbalancing:us-west-2:123456789012:listener-rule/
app/my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2/9683b2d02a6cabee
```

예제 2: 리스너에 대한 규칙을 설명하려면

다음 describe-rules 예제에서는 지정된 리스너의 규칙에 대한 세부 정보를 표시합니다. 출력에는 기본 규칙과 추가한 기타 규칙이 포함됩니다.

```
aws elbv2 describe-rules \
    --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/
my-load-balancer/50dc6c495c0c9188/f2f7dc8efc522ab2
```

- 자세한 API 내용은 명령 참조DescribeRules의 섹션을 참조하세요. AWS CLI

### describe-ssl-policies

다음 코드 예시에서는 describe-ssl-policies을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 협상에 사용되는 정책을 로드 SSL 밸런서 유형별로 나열하려면

다음 describe-ssl-policies 예제에서는 Application Load Balancer와의 SSL 협상에 사용할 수 있는 정책 이름을 표시합니다. 이 예제에서는 --query 파라미터를 사용하여 정책의 이름만 표시합니다.

```
aws elbv2 describe-ssl-policies \
    --load-balancer-type application \
    --query SslPolicies[*].Name
```

출력:

```
[
    "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"
]
```

예제 2: 특정 프로토콜을 지원하는 정책을 나열하려면

다음 `describe-ssl-policies` 예제에서는 TLS 1.3 프로토콜을 지원하는 정책의 이름을 표시합니다. 이 예제에서는 `--query` 파라미터를 사용하여 정책의 이름만 표시합니다.

```
aws elbv2 describe-ssl-policies \
    --load-balancer-type application \
    --query SslPolicies[?contains(SslProtocols,'TLSv1.3')].Name
```

출력:

```
[
    "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"
```

```
]
```

예제 3: 정책의 암호 표시

다음 `describe-ssl-policies` 예제에서는 지정된 정책의 암호 이름을 표시합니다. 이 예제에서는 `--query` 파라미터를 사용하여 암호 이름만 표시합니다. 목록의 첫 번째 암호는 우선 순위가 1이고 나머지 암호는 우선 순위가 입니다.

```
aws elbv2 describe-ssl-policies \
    --names ELBSecurityPolicy-TLS13-1-2-2021-06 \
    --query SslPolicies[*].Ciphers[*].Name
```

출력:

```
[
    "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"
]
```

자세한 내용은 Application Load Balancer 사용 설명서의 보안 정책을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeSslPolicies의 섹션을 참조하세요. AWS CLI

### `describe-tags`

다음 코드 예시에서는 `describe-tags`을 사용하는 방법을 보여 줍니다.

AWS CLI

로드 밸런서에 할당된 태그를 설명하려면

이 예제에서는 지정된 로드 밸런서에 할당된 태그를 설명합니다.

명령:

```
aws elbv2 describe-tags --resource-arns arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188
```

출력:

```
{
  "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"
              }
          ]
      }
  ]
}
```

- 자세한 API 내용은 명령 참조DescribeTags의 섹션을 참조하세요. AWS CLI

### describe-target-group-attributes

다음 코드 예시에서는 describe-target-group-attributes을 사용하는 방법을 보여 줍니다.

AWS CLI

대상 그룹 속성을 설명하려면

다음 describe-target-group-attributes 예제에서는 지정된 대상 그룹의 속성을 표시합니다.

```
aws elbv2 describe-target-group-attributes \
    --target-group-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

프로토콜이 HTTP 또는이고 대상 유형이 instance 또는 인 경우 출력HTTPS에 속성이 포함됩니다ip.

```
{
    "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"
        }
    ]
}
```

다음 출력에는 프로토콜이 HTTP 또는이고 HTTPS 대상 유형이 인 경우 속성이 포함됩니다lambda.

```
{
    "Attributes": [
        {
            "Value": "false",
            "Key": "lambda.multi_value_headers.enabled"
        }
    ]
}
```

다음 출력에는 프로토콜이 TCP, TLS, UDP또는 TCP_인 경우 속성이 포함됩니다UDP.

```
{
    "Attributes": [
        {
            "Value": "false",
            "Key": "proxy_protocol_v2.enabled"
        },
        {
            "Value": "300",
            "Key": "deregistration_delay.timeout_seconds"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeTargetGroupAttributes의 섹션을 참조하세요. AWS CLI

### `describe-target-groups`

다음 코드 예시에서는 `describe-target-groups`을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: 대상 그룹을 설명하는 방법

다음 `describe-target-groups` 예시에서는 지정된 대상 그룹의 세부 정보를 표시합니다.

```
aws elbv2 describe-target-groups \
    --target-group-arns arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

출력:

```
{
    "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"
        }
    ]
}
```

예 2: 로드 밸런서의 모든 대상 그룹을 설명하는 방법

다음 describe-target-groups 예시에서는 지정된 로드 밸런서의 모든 대상 그룹에 대한 세부 정보를 표시합니다. 이 예제에서는 --query 파라미터를 사용하여 대상 그룹 이름만 표시합니다.

```
aws elbv2 describe-target-groups \
    --load-balancer-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 \
    --query TargetGroups[*].TargetGroupName
```

출력:

```
[
    "my-instance-targets",
    "my-ip-targets",
    "my-lambda-target"
]
```

자세한 내용은 Applicaion Load Balancer 가이드의 대상 그룹을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeTargetGroups의 섹션을 참조하세요. AWS CLI

### describe-target-health

다음 코드 예시에서는 describe-target-health을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: 대상 그룹의 대상 상태를 설명하는 방법

다음 describe-target-health 예시에서는 지정된 대상 그룹의 대상 상태 세부 정보를 표시합니다. 이러한 대상은 정상입니다.

```
aws elbv2 describe-target-health \
    --target-group-arn arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
```

출력:

```
{
    "TargetHealthDescriptions": [
        {
            "HealthCheckPort": "80",
            "Target": {
                "Id": "i-ceddcd4d",
                "Port": 80
            },
            "TargetHealth": {
                "State": "healthy"
            }
        },
        {
            "HealthCheckPort": "80",
            "Target": {
                "Id": "i-0f76fade",
                "Port": 80
            },
            "TargetHealth": {
                "State": "healthy"
            }
        }
    ]
}
```

예 2: 대상의 상태를 설명하는 방법

다음 describe-target-health 예시에서는 지정된 대상의 상태 세부 정보를 표시합니다. 이 대상은 정상입니다.

```
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
```

출력:

```
{
    "TargetHealthDescriptions": [
        {
            "HealthCheckPort": "80",
            "Target": {
                "Id": "i-0f76fade",
                "Port": 80
            },
            "TargetHealth": {
                "State": "healthy"
            }
        }
    ]
}
```

다음 예시 출력은 리스너에 대한 작업에 대상 그룹이 지정되지 않은 대상에 대한 것입니다. 이 대상은 로드 밸런서에서 트래픽을 수신할 수 없습니다.

```
{
    "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"
            }
```

```
        }
    ]
}
```

다음 예시 출력은 리스너에 대한 작업에 대상 그룹이 방금 지정된 대상에 대한 것입니다. 대상이 아직 등록되는 중입니다.

```
{
    "TargetHealthDescriptions": [
        {
            "HealthCheckPort": "80",
            "Target": {
                "Id": "i-0f76fade",
                "Port": 80
            },
            "TargetHealth": {
                "State": "initial",
                "Reason": "Elb.RegistrationInProgress",
                "Description": "Target registration is in progress"
            }
        }
    ]
}
```

다음 예시 출력은 비정상 대상에 대한 것입니다.

```
{
    "TargetHealthDescriptions": [
        {
            "HealthCheckPort": "80",
            "Target": {
                "Id": "i-0f76fade",
                "Port": 80
            },
            "TargetHealth": {
                "State": "unhealthy",
                "Reason": "Target.Timeout",
                "Description": "Connection to target timed out"
            }
        }
    ]
}
```

다음 예시 출력은 Lambda 함수인 대상에 대한 것이며 상태 확인은 비활성화되어 있습니다.

```
{
    "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"
            }
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeTargetHealth의 섹션을 참조하세요. AWS CLI

### `modify-listener`

다음 코드 예시에서는 `modify-listener`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 기본 작업을 전달 작업으로 변경하려면

다음 `modify-listener` 예제에서는 지정된 리스너에 대한 기본 작업(전달 작업)을 변경합니다.

```
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
```

출력:

```
{
    "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"
        }
    ]
}
```

예제 2: 기본 작업을 리디렉션 작업으로 변경하려면

다음 `modify-listener` 예제에서는 기본 작업을 지정된 리스너에 대한 리디렉션 작업으로 변경합니다.

```
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
```

출력:

```
{
"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"
    }
  ]
}
```

예제 3: 서버 인증서 변경

이 예제에서는 지정된 HTTPS 리스너의 서버 인증서를 변경합니다.

```
aws elbv2 modify-listener \
    --listener-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/
my-load-balancer/50dc6c495c0c9188/0467ef3c8400ae65 \
    --certificates CertificateArn=arn:aws:iam::123456789012:server-certificate/my-
new-server-cert
```

출력:

```
{
    "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"
        }
```

```
    ]
}
```

- 자세한 API 내용은 명령 참조ModifyListener의 섹션을 참조하세요. AWS CLI

### `modify-load-balancer-attributes`

다음 코드 예시에서는 `modify-load-balancer-attributes`을 사용하는 방법을 보여 줍니다.

AWS CLI

삭제 방지를 활성화하려면

이 예제에서는 지정된 로드 밸런서에 대한 삭제 보호를 활성화합니다.

명령:

```
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
```

출력:

```
{
  "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"
        }
    ]
}
```

유휴 제한 시간을 변경하려면

이 예제에서는 지정된 로드 밸런서에 대한 유휴 제한 시간 값을 변경합니다.

명령:

```
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
```

출력:

```
{
    "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"
        }
    ]
}
```

액세스 로그를 활성화하려면

이 예제에서는 지정된 로드 밸런서에 대한 액세스 로그를 활성화합니다. S3 버킷은 로드 밸런서와 동일한 리전에 있어야 하며 Elastic Load Balancing 서비스에 대한 액세스 권한을 부여하는 정책이 연결되어 있어야 합니다.

명령:

```
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
```

출력:

```
{
  "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"
      }
  ]
}
```

- 자세한 API 내용은 명령 참조ModifyLoadBalancerAttributes의 섹션을 참조하세요. AWS CLI

### modify-rule

다음 코드 예시에서는 modify-rule을 사용하는 방법을 보여 줍니다.

AWS CLI

규칙을 수정하려면

다음 modify-rule 예제에서는 지정된 규칙에 대한 작업 및 조건을 업데이트합니다.

```
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
```

출력:

```
{
    "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"
                }
            ]
        }
```

```
    ]
}
```

- 자세한 API 내용은 명령 참조ModifyRule의 섹션을 참조하세요. AWS CLI

### `modify-target-group-attributes`

다음 코드 예시에서는 `modify-target-group-attributes`을 사용하는 방법을 보여 줍니다.

AWS CLI

등록 취소 지연 제한 시간을 수정하려면

이 예제에서는 등록 취소 지연 제한 시간을 지정된 대상 그룹에 지정된 값으로 설정합니다.

명령:

```
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
```

출력:

```
{
  "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"
      }
```

```
  ]
}
```

- 자세한 API 내용은 명령 참조ModifyTargetGroupAttributes의 섹션을 참조하세요. AWS CLI

### modify-target-group

다음 코드 예시에서는 modify-target-group을 사용하는 방법을 보여 줍니다.

AWS CLI

대상 그룹의 상태 확인 구성을 수정하려면

다음 modify-target-group 예제에서는 지정된 대상 그룹의 대상 상태를 평가하는 데 사용되는 상태 확인의 구성을 변경합니다. 쉼표CLI를 구문 분석하는 방식으로 인해 --matcher 옵션의 범위를 큰따옴표 대신 작은따옴표로 둘러싸야 합니다.

```
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'
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Applicaion Load Balancer 가이드의 대상 그룹을 참조하세요.

- 자세한 API 내용은 명령 참조ModifyTargetGroup의 섹션을 참조하세요. AWS CLI

### register-targets

다음 코드 예시에서는 register-targets을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 인스턴스 ID별로 대상 그룹에 대상을 등록하려면

다음 register-targets 예제에서는 지정된 인스턴스를 대상 그룹에 등록합니다. 대상 그룹의 대상 유형은 여야 합니다instance.

```
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
```

예제 2: 포트 재정의를 사용하여 대상 그룹에 대상을 등록하려면

다음 register-targets 예제에서는 여러 포트를 사용하여 지정된 인스턴스를 대상 그룹에 등록합니다. 이렇게 하면 대상 그룹의 대상과 동일한 인스턴스에 컨테이너를 등록할 수 있습니다.

```
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
```

예제 3: IP 주소별로 대상 그룹에 대상을 등록하려면

다음 `register-targets` 예제에서는 지정된 IP 주소를 대상 그룹에 등록합니다. 대상 그룹의 대상 유형은 이어야 합니다`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
```

예제 4: Lambda 함수를 대상으로 등록하려면

다음 `register-targets` 예제에서는 지정된 IP 주소를 대상 그룹에 등록합니다. 대상 그룹의 대상 유형은 여야 합니다`lambda`. Lambda 함수를 호출하려면 Elastic Load Balancing 권한을 부여해야 합니다.

```
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
```

- 자세한 API 내용은 명령 참조RegisterTargets의 섹션을 참조하세요. AWS CLI

### `remove-listener-certificates`

다음 코드 예시에서는 `remove-listener-certificates`을 사용하는 방법을 보여 줍니다.

AWS CLI

보안 리스너에서 인증서를 제거하려면

이 예제에서는 지정된 보안 리스너에서 지정된 인증서를 제거합니다.

명령:

```
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
```

- 자세한 API 내용은 명령 참조RemoveListenerCertificates의 섹션을 참조하세요. AWS CLI

### remove-tags

다음 코드 예시에서는 remove-tags을 사용하는 방법을 보여 줍니다.

AWS CLI

로드 밸런서에서 태그를 제거하려면

다음 remove-tags 예제에서는 지정된 로드 밸런서에서 project 및 department 태그를 제거합니다.

```
aws elbv2 remove-tags \
    --resource-arns arn:aws:elasticloadbalancing:us-
west-2:123456789012:loadbalancer/app/my-load-balancer/50dc6c495c0c9188 \
    --tag-keys project department
```

- 자세한 API 내용은 명령 참조RemoveTags의 섹션을 참조하세요. AWS CLI

### set-ip-address-type

다음 코드 예시에서는 set-ip-address-type을 사용하는 방법을 보여 줍니다.

AWS CLI

로드 밸런서의 주소 유형을 설정하려면

이 예제에서는 지정된 로드 밸런서의 주소 유형을 로 설정합니다dualstack. 로드 밸런서 서브넷에는 연결된 IPv6 CIDR 블록이 있어야 합니다.

명령:

```
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
```

출력:

```
{
    "IpAddressType": "dualstack"
}
```

- 자세한 API 내용은 명령 참조SetIpAddressType의 섹션을 참조하세요. AWS CLI

### set-rule-priorities

다음 코드 예시에서는 set-rule-priorities을 사용하는 방법을 보여 줍니다.

AWS CLI

규칙 우선 순위를 설정하려면

이 예제에서는 지정된 규칙의 우선 순위를 설정합니다.

명령:

```
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
```

출력:

```
{
  "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"
            }
        ]
    }
  ]
}
```

- 자세한 API 내용은 명령 참조SetRulePriorities의 섹션을 참조하세요. AWS CLI

### set-security-groups

다음 코드 예시에서는 set-security-groups을 사용하는 방법을 보여 줍니다.

AWS CLI

보안 그룹을 로드 밸런서에 연결하려면

이 예제에서는 지정된 보안 그룹을 지정된 로드 밸런서에 연결합니다.

명령:

```
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
```

출력:

```
{
  "SecurityGroupIds": [
      "sg-5943793c"
  ]
}
```

- 자세한 API 내용은 명령 참조SetSecurityGroups의 섹션을 참조하세요. AWS CLI

### set-subnets

다음 코드 예시에서는 set-subnets을 사용하는 방법을 보여 줍니다.

AWS CLI

로드 밸런서에 대해 가용 영역을 활성화하려면

이 예제에서는 지정된 로드 밸런서에 대해 지정된 서브넷의 가용 영역을 활성화합니다.

명령:

```
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
```

출력:

```
{
  "AvailabilityZones": [
      {
          "SubnetId": "subnet-8360a9e7",
          "ZoneName": "us-west-2a"
      },
      {
          "SubnetId": "subnet-b7d581c0",
          "ZoneName": "us-west-2b"
      }
  ]
}
```

- 자세한 API 내용은 명령 참조SetSubnets의 섹션을 참조하세요. AWS CLI

## 를 사용한 Elastic Transcoder 예제 AWS CLI

다음 코드 예제에서는 Elastic Transcoder AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### cancel-job

다음 코드 예시에서는 cancel-job을 사용하는 방법을 보여 줍니다.

AWS CLI

에 대한 작업을 취소하려면 ElasticTranscoder

이렇게 하면 에 대해 지정된 작업이 취소됩니다 ElasticTranscoder.

명령:

```
aws elastictranscoder cancel-job --id 3333333333333-abcde3
```

- 자세한 API 내용은 명령 참조CancelJob의 섹션을 참조하세요. AWS CLI

### create-job

다음 코드 예시에서는 create-job을 사용하는 방법을 보여 줍니다.

AWS CLI

에 대한 작업을 생성하려면 ElasticTranscoder

다음 create-job 예제에서는 에 대한 작업을 생성합니다 ElasticTranscoder.

```
aws elastictranscoder create-job \
    --pipeline-id 1111111111111-abcde1 \
    --inputs file://inputs.json \
    --outputs file://outputs.json \
    --output-key-prefix "recipes/" \
    --user-metadata file://user-metadata.json
```

inputs.json의 콘텐츠:

```
[{
    "Key":"ETS_example_file.mp4",
    "FrameRate":"auto",
```

```
    "Resolution":"auto",
    "AspectRatio":"auto",
    "Interlaced":"auto",
    "Container":"mp4"
}]
```

outputs.json의 내용:

```
[
    {
        "Key":"webm/ETS_example_file-kindlefirehd.webm",
        "Rotate":"0",
        "PresetId":"1351620000001-100250"
    }
]
```

user-metadata.json의 콘텐츠:

```
{
    "Food type":"Italian",
    "Cook book":"recipe notebook"
}
```

출력:

```
{
    "Job": {
        "Status": "Submitted",
        "Inputs": [
            {
                "Container": "mp4",
                "FrameRate": "auto",
                "Key": "ETS_example_file.mp4",
                "AspectRatio": "auto",
                "Resolution": "auto",
                "Interlaced": "auto"
            }
        ],
        "Playlists": [],
        "Outputs": [
            {
                "Status": "Submitted",
```

```
                "Rotate": "0",
                "PresetId": "1351620000001-100250",
                "Watermarks": [],
                "Key": "webm/ETS_example_file-kindlefirehd.webm",
                "Id": "1"
            }
        ],
        "PipelineId": "3333333333333-abcde3",
        "OutputKeyPrefix": "recipes/",
        "UserMetadata": {
            "Cook book": "recipe notebook",
            "Food type": "Italian"
        },
        "Output": {
            "Status": "Submitted",
            "Rotate": "0",
            "PresetId": "1351620000001-100250",
            "Watermarks": [],
            "Key": "webm/ETS_example_file-kindlefirehd.webm",
            "Id": "1"
        },
        "Timing": {
            "SubmitTimeMillis": 1533838012298
        },
        "Input": {
            "Container": "mp4",
            "FrameRate": "auto",
            "Key": "ETS_example_file.mp4",
            "AspectRatio": "auto",
            "Resolution": "auto",
            "Interlaced": "auto"
        },
        "Id": "1533838012294-example",
        "Arn": "arn:aws:elastictranscoder:us-west-2:123456789012:job/1533838012294-
example"
    }
}
```

- 자세한 API 내용은 명령 참조CreateJob의 섹션을 참조하세요. AWS CLI

### `create-pipeline`

다음 코드 예시에서는 `create-pipeline`을 사용하는 방법을 보여 줍니다.

AWS CLI

에 대한 파이프라인을 생성하려면 ElasticTranscoder

다음 `create-pipeline` 예제에서는 에 대한 파이프라인을 생성합니다 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
```

`content-config.json`의 콘텐츠:

```
{
    "Bucket":"salesoffice.example.com-public-promos",
    "Permissions":[
        {
            "GranteeType":"Email",
            "Grantee":"marketing-promos@example.com",
            "Access":[
            "FullControl"
            ]
        }
    ],
    "StorageClass":"Standard"
}
```

`thumbnail-config.json`의 콘텐츠:

```
{
    "Bucket":"salesoffice.example.com-public-promos-thumbnails",
    "Permissions":[
        {
            "GranteeType":"Email",
            "Grantee":"marketing-promos@example.com",
            "Access":[
                "FullControl"
            ]
```

```
        }
    ],
    "StorageClass":"ReducedRedundancy"
}
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조CreatePipeline의 섹션을 참조하세요. AWS CLI

### `create-preset`

다음 코드 예시에서는 `create-preset`을 사용하는 방법을 보여 줍니다.

AWS CLI

에 대한 사전 설정을 생성하려면 ElasticTranscoder

다음 `create-preset` 예제에서는 에 대한 사전 설정을 생성합니다 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
```

`video.json`의 콘텐츠:

```
{
    "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"
        }
    ]
}
```

audio.json의 콘텐츠:

```
{
    "Codec":"AAC",
    "CodecOptions":{
        "Profile":"AAC-LC"
    },
    "SampleRate":"44100",
```

```
    "BitRate":"96",
    "Channels":"2"
}
```

thumbnails.json의 콘텐츠:

```
{
    "Format":"png",
    "Interval":"120",
    "MaxWidth":"auto",
    "MaxHeight":"auto",
    "SizingPolicy":"Fit",
    "PaddingPolicy":"Pad"
}
```

출력:

```
{
    "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": ""
}
```

- 자세한 API 내용은 명령 참조CreatePreset의 섹션을 참조하세요. AWS CLI

### delete-pipeline

다음 코드 예시에서는 delete-pipeline을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 ElasticTranscoder 파이프라인을 삭제하려면

이렇게 하면 지정된 ElasticTranscoder 파이프라인이 삭제됩니다.

명령:

```
aws elastictranscoder delete-pipeline --id 1111111111111-abcde1
```

출력:

```
{
 "Success":"true"
}
```

- 자세한 API 내용은 명령 참조DeletePipeline의 섹션을 참조하세요. AWS CLI

### delete-preset

다음 코드 예시에서는 delete-preset을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 ElasticTranscoder 사전 설정을 삭제하려면

이렇게 하면 지정된 ElasticTranscoder 사전 설정이 삭제됩니다.

명령:

```
aws elastictranscoder delete-preset --id 5555555555555-abcde5
```

- 자세한 API 내용은 명령 참조DeletePreset의 섹션을 참조하세요. AWS CLI

### list-jobs-by-pipeline

다음 코드 예시에서는 list-jobs-by-pipeline을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 파이프라인에서 ElasticTranscoder 작업 목록을 검색하려면

이 예제에서는 지정된 파이프라인의 ElasticTranscoder 작업 목록을 검색합니다.

명령:

```
aws elastictranscoder list-jobs-by-pipeline --pipeline-id 1111111111111-abcde1
```

출력:

```
{
   "Jobs": []
}
```

- 자세한 API 내용은 명령 참조ListJobsByPipeline의 섹션을 참조하세요. AWS CLI

### `list-jobs-by-status`

다음 코드 예시에서는 `list-jobs-by-status`을 사용하는 방법을 보여 줍니다.

AWS CLI

상태가 완료인 ElasticTranscoder 작업 목록을 검색하려면

이 예제에서는 상태가 완료인 ElasticTranscoder 작업 목록을 검색합니다.

명령:

```
aws elastictranscoder list-jobs-by-status --status Complete
```

출력:

```
{
   "Jobs": []
}
```

- 자세한 API 내용은 명령 참조ListJobsByStatus의 섹션을 참조하세요. AWS CLI

### list-pipelines

다음 코드 예시에서는 list-pipelines을 사용하는 방법을 보여 줍니다.

AWS CLI

ElasticTranscoder 파이프라인 목록을 검색하려면

이 예제에서는 ElasticTranscoder 파이프라인 목록을 검색합니다.

명령:

```
aws elastictranscoder list-pipelines
```

출력:

```
{
   "Pipelines": [
       {
           "Status": "Active",
           "ContentConfig": {
               "Bucket": "ets-example",
               "Permissions": []
           },
           "Name": "example-pipeline",
           "ThumbnailConfig": {
               "Bucket": "ets-example",
               "Permissions": []
           },
           "Notifications": {
               "Completed": "arn:aws:sns:us-west-2:123456789012:ets_example",
               "Warning": "",
               "Progressing": "",
               "Error": ""
           },
           "Role": "arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
           "InputBucket": "ets-example",
           "OutputBucket": "ets-example",
           "Id": "3333333333333-abcde3",
           "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:pipeline/3333333333333-abcde3"
       },
       {
           "Status": "Paused",
```

```
            "ContentConfig": {
                "Bucket": "ets-example",
                "Permissions": []
            },
            "Name": "example-php-test",
            "ThumbnailConfig": {
                "Bucket": "ets-example",
                "Permissions": []
            },
            "Notifications": {
                "Completed": "",
                "Warning": "",
                "Progressing": "",
                "Error": ""
            },
            "Role": "arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
            "InputBucket": "ets-example",
            "OutputBucket": "ets-example",
            "Id": "3333333333333-abcde2",
            "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:pipeline/3333333333333-abcde2"
        },
        {
            "Status": "Active",
            "ContentConfig": {
                "Bucket": "ets-west-output",
                "Permissions": []
            },
            "Name": "pipeline-west",
            "ThumbnailConfig": {
                "Bucket": "ets-west-output",
                "Permissions": []
            },
            "Notifications": {
                "Completed": "arn:aws:sns:us-west-2:123456789012:ets-notifications",
                "Warning": "",
                "Progressing": "",
                "Error": ""
            },
            "Role": "arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
            "InputBucket": "ets-west-input",
            "OutputBucket": "ets-west-output",
            "Id": "3333333333333-abcde1",
```

```
            "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:pipeline/3333333333333-abcde1"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListPipelines의 섹션을 참조하세요. AWS CLI

### `list-presets`

다음 코드 예시에서는 `list-presets`을 사용하는 방법을 보여 줍니다.

AWS CLI

ElasticTranscoder 사전 설정 목록을 검색하려면

이 예제에서는 ElasticTranscoder 사전 설정 목록을 검색합니다.

명령:

```
aws elastictranscoder list-presets --max-items 2
```

출력:

```
{
  "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": "3333333333333-abcde2",
        "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:preset/3333333333333-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": "3333333333333-abcde3",
        "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:preset/3333333333333-abcde3",
        "Name": "Roman's Preset"
    }
  ],
  "NextToken": "eyJQYWdlVG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ=="
}
```

- 자세한 API 내용은 명령 참조ListPresets의 섹션을 참조하세요. AWS CLI

### read-job

다음 코드 예시에서는 read-job을 사용하는 방법을 보여 줍니다.

AWS CLI

ElasticTranscoder 작업을 검색하려면

이 예제에서는 지정된 ElasticTranscoder 작업을 검색합니다.

명령:

```
aws elastictranscoder read-job --id 1533838012294-example
```

출력:

```
{
```

```
    "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"
    }
}
```

- 자세한 API 내용은 명령 참조ReadJob의 섹션을 참조하세요. AWS CLI

### `read-pipeline`

다음 코드 예시에서는 `read-pipeline`을 사용하는 방법을 보여 줍니다.

AWS CLI

ElasticTranscoder 파이프라인을 검색하려면

이 예제에서는 지정된 ElasticTranscoder 파이프라인을 검색합니다.

명령:

```
aws elastictranscoder read-pipeline --id 3333333333333-abcde3
```

출력:

```
{
    "Pipeline": {
        "Status": "Active",
        "ContentConfig": {
            "Bucket": "ets-example",
            "StorageClass": "Standard",
            "Permissions": [
                {
                    "Access": [
                        "FullControl"
                    ],
                    "Grantee": "marketing-promos@example.com",
```

```
                    "GranteeType": "Email"
                }
            ]
        },
        "Name": "Default",
        "ThumbnailConfig": {
            "Bucket": "ets-example",
            "StorageClass": "ReducedRedundancy",
            "Permissions": [
                {
                    "Access": [
                        "FullControl"
                    ],
                    "Grantee": "marketing-promos@example.com",
                    "GranteeType": "Email"
                }
            ]
        },
        "Notifications": {
            "Completed": "",
            "Warning": "",
            "Progressing": "",
            "Error": "arn:aws:sns:us-east-1:123456789012:ETS_Errors"
        },
        "Role": "arn:aws:iam::123456789012:role/Elastic_Transcoder_Default_Role",
        "InputBucket": "ets-example",
        "Id": "3333333333333-abcde3",
        "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:pipeline/3333333333333-abcde3"
    },
    "Warnings": [
        {
            "Message": "The SNS notification topic for Error events and the pipeline
 are in different regions, which increases processing time for jobs in the pipeline
 and can incur additional charges. To decrease processing time and prevent cross-
regional charges, use the same region for the SNS notification topic and the
 pipeline.",
            "Code": "6006"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ReadPipeline의 섹션을 참조하세요. AWS CLI

### read-preset

다음 코드 예시에서는 read-preset을 사용하는 방법을 보여 줍니다.

AWS CLI

ElasticTranscoder 사전 설정을 검색하려면

이 예제에서는 지정된 ElasticTranscoder 사전 설정을 검색합니다.

명령:

```
aws elastictranscoder read-preset --id 1351620000001-500020
```

출력:

```
{
   "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"
    }
}
```

- 자세한 API 내용은 명령 참조ReadPreset의 섹션을 참조하세요. AWS CLI

### update-pipeline-notifications

다음 코드 예시에서는 update-pipeline-notifications을 사용하는 방법을 보여 줍니다.

AWS CLI

ElasticTranscoder 파이프라인 알림을 업데이트하려면

이 예제에서는 지정된 ElasticTranscoder 파이프라인의 알림을 업데이트합니다.

명령:

```
aws elastictranscoder update-pipeline-notifications --id 1111111111111-
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
```

출력:

```
{
```

```
    "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": "1111111111111-abcde1",
        "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:pipeline/1111111111111-abcde1"
    }
}
```

- 자세한 API 내용은 명령 참조UpdatePipelineNotifications의 섹션을 참조하세요. AWS CLI

### update-pipeline-status

다음 코드 예시에서는 update-pipeline-status을 사용하는 방법을 보여 줍니다.

AWS CLI

ElasticTranscoder 파이프라인의 상태를 업데이트하려면

이 예제에서는 지정된 ElasticTranscoder 파이프라인의 상태를 업데이트합니다.

명령:

```
aws elastictranscoder update-pipeline-status --id 1111111111111-abcde1 --
status Paused
```

출력:

```
{
   "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": "1111111111111-abcde1",
        "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:pipeline/1111111111111-abcde1"
    }
}
```

- 자세한 API 내용은 명령 참조UpdatePipelineStatus의 섹션을 참조하세요. AWS CLI

### update-pipeline

다음 코드 예시에서는 update-pipeline을 사용하는 방법을 보여 줍니다.

AWS CLI

ElasticTranscoder 파이프라인을 업데이트하려면

다음 update-pipeline 예제에서는 지정된 ElasticTranscoder 파이프라인을 업데이트합니다.

```
aws elastictranscoder update-pipeline \
    --id 1111111111111-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
```

content-config.json의 콘텐츠:

```
{
```

```
    "Bucket":"salesoffice.example.com-public-promos",
    "Permissions":[
        {
            "GranteeType":"Email",
            "Grantee":"marketing-promos@example.com",
            "Access":[
                "FullControl"
            ]
        }
    ],
    "StorageClass":"Standard"
}
```

thumbnail-config.json의 콘텐츠:

```
{
    "Bucket":"salesoffice.example.com-public-promos-thumbnails",
    "Permissions":[
        {
            "GranteeType":"Email",
            "Grantee":"marketing-promos@example.com",
            "Access":[
                "FullControl"
            ]
        }
    ],
    "StorageClass":"ReducedRedundancy"
}
```

출력:

```
{
    "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": "3333333333333-abcde3",
        "Arn": "arn:aws:elastictranscoder:us-
west-2:123456789012:pipeline/3333333333333-abcde3"
    },
    "Warnings": [
        {
            "Message": "The SNS notification topic for Error events and the pipeline
 are in different regions, which increases processing time for jobs in the pipeline
 and can incur additional charges. To decrease processing time and prevent cross-
regional charges, use the same region for the SNS notification topic and the
 pipeline.",
            "Code": "6006"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조[UpdatePipeline](#)의 섹션을 참조하세요. AWS CLI

# ElastiCache 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 ElastiCache.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `add-tags-to-resource`

다음 코드 예시에서는 `add-tags-to-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 태그를 추가하려면

다음 `add-tags-to-resource` 예제에서는 클러스터 또는 스냅샷 리소스에 최대 10개의 태그, 키-값 페어를 추가합니다.

```
aws elasticache add-tags-to-resource \
    --resource-name "arn:aws:elasticache:us-east-1:1234567890:cluster:my-mem-cluster" \
    --tags '{"20150202":15, "ElastiCache":"Service"}'
```

출력:

```
{
    "TagList": [
        {
            "Value": "20150202",
            "Key": "APIVersion"
        },
```

```
        {
            "Value": "ElastiCache",
            "Key": "Service"
        }
    ]
}
```

자세한 내용은 Elasticache 사용 설명서의 비용 할당 태그를 사용한 비용 모니터링을 참조하세요.

- 자세한 API 내용은 명령 참조AddTagsToResource의 섹션을 참조하세요. AWS CLI

### `authorize-cache-security-group-ingress`

다음 코드 예시에서는 `authorize-cache-security-group-ingress`을 사용하는 방법을 보여 줍니다.

AWS CLI

캐시 보안 그룹에 수신을 승인하려면

다음 `authorize-cache-security-group-ingress` 예제에서는 캐시 보안 그룹에 대한 네트워크 수신을 허용합니다.

```
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"
```

명령은 출력을 생성하지 않습니다.

자세한 내용은 Elasticache 사용 설명서의 Amazon에서 셀프 서비스 업데이트를 참조하세요 ElastiCache.

- 자세한 API 내용은 명령 참조AuthorizeCacheSecurityGroupIngress의 섹션을 참조하세요. AWS CLI

### `batch-apply-update-action`

다음 코드 예시에서는 `batch-apply-update-action`을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스 업데이트를 적용하려면

다음 batch-apply-update-action 예제에서는 서비스 업데이트를 Redis 클러스터에 적용합니다.

```
aws elasticache batch-apply-update-action \
    --service-update-name elc-xxxxx406-xxx \
    --replication-group-ids test-cluster
```

출력:

```
{
    "ProcessedUpdateActions": [
        {
            "ReplicationGroupId": "pat-cluster",
            "ServiceUpdateName": "elc-xxxxx406-xxx",
            "UpdateActionStatus": "waiting-to-start"
        }
    ],
    "UnprocessedUpdateActions": []
}
```

자세한 내용은 Elasticache 사용 설명서의 Amazon에서 셀프 서비스 업데이트를 참조하세요 ElastiCache.

- 자세한 API 내용은 명령 참조BatchApplyUpdateAction의 섹션을 참조하세요. AWS CLI

### batch-stop-update-action

다음 코드 예시에서는 batch-stop-update-action을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스 업데이트를 중지하려면

다음 batch-stop-update-action 예제에서는 서비스 업데이트를 Redis 클러스터에 적용합니다.

```
aws elasticache batch-stop-update-action \
    --service-update-name elc-xxxxx406-xxx \
    --replication-group-ids test-cluster
```

출력:

```
{
    "ProcessedUpdateActions": [
        {
            "ReplicationGroupId": "pat-cluster",
            "ServiceUpdateName": "elc-xxxxx406-xxx",
            "UpdateActionStatus": "stopping"
        }
    ],
    "UnprocessedUpdateActions": []
}
```

자세한 내용은 Elasticache 사용 설명서의 Amazon에서 셀프 서비스 업데이트를 참조하세요 ElastiCache.

- 자세한 API 내용은 명령 참조BatchStopUpdateAction의 섹션을 참조하세요. AWS CLI

### copy-snapshot

다음 코드 예시에서는 copy-snapshot을 사용하는 방법을 보여 줍니다.

AWS CLI

스냅샷을 복사하려면

다음 copy-snapshot 예제에서는 기존 스냅샷의 복사본을 만듭니다.

```
aws elasticache copy-snapshot \
    --source-snapshot-name "my-snapshot" \
    --target-snapshot-name "my-snapshot-copy"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Elasticache 사용 설명서의 백업 내보내기를 참조하세요.

- 자세한 API 내용은 명령 참조CopySnapshot의 섹션을 참조하세요. AWS CLI

### `create-cache-cluster`

다음 코드 예시에서는 `create-cache-cluster`을 사용하는 방법을 보여 줍니다.

AWS CLI

캐시 클러스터를 생성하려면

다음 `create-cache-cluster` 예제에서는 Redis 엔진을 사용하여 캐시 클러스터를 생성합니다.

```
aws elasticache create-cache-cluster \
    --cache-cluster-id "cluster-test" \
    --engine redis \
    --cache-node-type cache.m5.large \
    --num-cache-nodes 1
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 Elasticache 사용 설명서의 클러스터 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateCacheCluster의 섹션을 참조하세요. AWS CLI

### `create-cache-parameter-group`

다음 코드 예시에서는 `create-cache-parameter-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

캐시 파라미터 그룹을 생성하려면

다음 `create-cache-parameter-group` 예제에서는 새 Amazon ElastiCache 캐시 파라미터 그룹을 생성합니다.

```
aws elasticache create-cache-parameter-group \
```

```
    --cache-parameter-group-family "redis5.0" \
    --cache-parameter-group-name "mygroup" \
    --description "mygroup"
```

출력:

```
{
    "CacheParameterGroup": {
        "CacheParameterGroupName": "mygroup",
        "CacheParameterGroupFamily": "redis5.0",
        "Description": "my group"
    }
}
```

자세한 내용은 Elasticache 사용 설명서의 파라미터 그룹 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateCacheParameterGroup의 섹션을 참조하세요. AWS CLI

### create-cache-subnet-group

다음 코드 예시에서는 create-cache-subnet-group을 사용하는 방법을 보여 줍니다.

AWS CLI

캐시 서브넷 그룹을 생성하려면

다음 create-cache-subnet-group 예제에서는 새 캐시 서브넷 그룹을 생성합니다.

```
aws elasticache create-cache-subnet-group \
    --cache-subnet-group-name "mygroup" \
    --cache-subnet-group-description "my subnet group" \
    --subnet-ids "subnet-xxxxec4f"
```

출력:

```
{
    "CacheSubnetGroup": {
        "CacheSubnetGroupName": "mygroup",
        "CacheSubnetGroupDescription": "my subnet group",
        "VpcId": "vpc-a3e97cdb",
        "Subnets": [
            {
```

```
                "SubnetIdentifier": "subnet-xxxxec4f",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2d"
                }
            }
        ]
    }
}
```

자세한 내용은 Elasticache 사용 설명서의 캐시 서브넷 그룹 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateCacheSubnetGroup의 섹션을 참조하세요. AWS CLI

### `create-global-replication-group`

다음 코드 예시에서는 `create-global-replication-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

전역 복제 그룹을 생성하려면

다음 `create-global-replication-group` 예제에서는 새 전역 복제 그룹을 생성합니다.

```
aws elasticache create-global-replication-group \
    --global-replication-group-id-suffix my-global-replication-group \
    --primary-replication-group-id my-primary-cluster
```

출력:

```
{
    "GlobalReplicationGroup": {
        "GlobalReplicationGroupId": "sgaui-my-global-replication-group",
        "GlobalReplicationGroupDescription": " ",
        "Status": "creating",
        "CacheNodeType": "cache.r5.large",
        "Engine": "redis",
        "EngineVersion": "5.0.6",
        "Members": [
            {
                "ReplicationGroupId": "my-primary-cluster",
                "ReplicationGroupRegion": "us-west-2",
                "Role": "PRIMARY",
```

```
                "AutomaticFailover": "enabled",
                "Status": "associating"
            }
        ],
        "ClusterEnabled": true,
        "GlobalNodeGroups": [
            {
                "GlobalNodeGroupId": "sgaui-my-global-replication-group-0001",
                "Slots": "0-16383"
            }
        ],
        "AuthTokenEnabled": false,
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
}
```

자세한 내용은 Elasticache 사용 설명서의 글로벌 데이터 스토어를 사용하여 AWS 리전 간 복제를 참조하세요.

- 자세한 API 내용은 명령 참조CreateGlobalReplicationGroup의 섹션을 참조하세요. AWS CLI

### `create-replication-group`

다음 코드 예시에서는 `create-replication-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

복제 그룹을 생성하려면

다음 `create-replication-group` 예제에서는 Redis(클러스터 모드 비활성화됨) 또는 Redis(클러스터 모드 활성화됨) 복제 그룹을 생성합니다. 이 작업은 Redis에만 유효합니다.

```
aws elasticache create-replication-group \
    --replication-group-id "mygroup" \
    --replication-group-description "my group" \
    --engine "redis" \
    --cache-node-type "cache.m5.large"
```

출력:

```
{
```

```
    "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
    }
}
```

자세한 내용은 Elasticache 사용 설명서의 Redis 복제 그룹 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateReplicationGroup의 섹션을 참조하세요. AWS CLI

### `create-snapshot`

다음 코드 예시에서는 `create-snapshot`을 사용하는 방법을 보여 줍니다.

AWS CLI

스냅샷을 생성하려면

다음 `create-snapshot` 예제에서는 Redis 엔진을 사용하여 스냅샷을 생성합니다.

```
aws elasticache create-snapshot \
    --snapshot-name mysnapshot \
    --cache-cluster-id cluster-test
```

출력:

```
{
    "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"
            }
        ]
    }
}
```

자세한 내용은 Elasticache 사용 설명서의 Redis ElastiCache 용 백업 및 복원을 참조하세요.

- 자세한 API 내용은 명령 참조CreateSnapshot의 섹션을 참조하세요. AWS CLI

### `create-user-group`

다음 코드 예시에서는 `create-user-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 그룹을 생성하려면

다음 `create-user-group` 예제에서는 새 사용자 그룹을 생성합니다.

```
aws elasticache create-user-group \
    --user-group-id myusergroup \
    --engine redis \
    --user-ids default
```

출력:

```
{
    "UserGroupId": "myusergroup",
    "Status": "creating",
    "Engine": "redis",
    "UserIds": [
        "default"
    ],
    "ReplicationGroups": [],
    "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxx52:usergroup:myusergroup"
}
```

자세한 내용은 Elasticache 사용 설명서의 역할 기반 액세스 제어(RBAC)를 사용하여 사용자 인증을 참조하세요.

- 자세한 API 내용은 명령 참조CreateUserGroup의 섹션을 참조하세요. AWS CLI

### `create-user`

다음 코드 예시에서는 `create-user`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자를 생성하려면

다음 `create-user` 예제에서는 새 사용자를 생성합니다.

```
aws elasticache create-user \
    --user-id user1 \
    --user-name myUser \
    --passwords mYnuUzrpAxXw2rdzx \
    --engine redis \
    --access-string "on ~app::* -@all +@read"
```

출력:

```
{
    "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"
}
```

자세한 내용은 Elasticache 사용 설명서의 역할 기반 액세스 제어(RBAC)를 사용하여 사용자 인증을 참조하세요.

- 자세한 API 내용은 명령 참조CreateUser의 섹션을 참조하세요. AWS CLI

### `decrease-node-groups-in-global-replication-group`

다음 코드 예시에서는 `decrease-node-groups-in-global-replication-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

전역 복제 그룹의 노드 그룹 수를 줄이려면

다음은 Redis 엔진을 사용하여 노드 그룹 수를 `decrease-node-groups-in-global-replication-group` 줄입니다.

```
aws elasticache decrease-node-groups-in-global-replication-group \
    --global-replication-group-id sgaui-test \
    --node-group-count 1 \
    --apply-immediately \
    --global-node-groups-to-retain sgaui-test-0003
```

출력:

```
{
    "GlobalReplicationGroup":
    {
        "GlobalReplicationGroupId": "sgaui-test",
        "GlobalReplicationGroupDescription": "test",
        "Status": "modifying",
        "CacheNodeType": "cache.r5.large",
```

```
        "Engine": "redis",
        "EngineVersion": "5.0.6",
        "Members": [
            {
                "ReplicationGroupId": "test-2",
                "ReplicationGroupRegion": "us-east-1",
                "Role": "SECONDARY",
                "AutomaticFailover": "enabled",
                "Status": "associated"
            },
            {
                "ReplicationGroupId": "test-1",
                "ReplicationGroupRegion": "us-west-2",
                "Role": "PRIMARY",
                "AutomaticFailover": "enabled",
                "Status": "associated"
            }
        ],
        "ClusterEnabled": true,
        "GlobalNodeGroups": [
            {
                "GlobalNodeGroupId": "sgaui-test-0001",
                "Slots": "0-449,1816-5461"
            },
            {
                "GlobalNodeGroupId": "sgaui-test-0002",
                "Slots": "6827-10922"
            },
            {
                "GlobalNodeGroupId": "sgaui-test-0003",
                "Slots": "10923-14052,15418-16383"
            },
            {
                "GlobalNodeGroupId": "sgaui-test-0004",
                "Slots": "450-1815,5462-6826,14053-15417"
            }
        ],
        "AuthTokenEnabled": false,
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
}
```

자세한 내용은 Elasticache 사용 설명서의 글로벌 데이터 스토어를 사용하여 AWS 리전 간 복제를 참조하세요.

- 자세한 API 내용은 명령 참조DecreaseNodeGroupsInGlobalReplicationGroup의 섹션을 참조하세요. AWS CLI

### decrease-replica-count

다음 코드 예시에서는 decrease-replica-count을 사용하는 방법을 보여 줍니다.

AWS CLI

복제본 수를 줄이려면

다음 decrease-replica-count 예제에서는 Redis(클러스터 모드 비활성화됨) 복제 그룹의 복제본 수 또는 Redis(클러스터 모드 활성화됨) 복제 그룹의 하나 이상의 노드 그룹(샤드)에 있는 복제본 노드 수를 동적으로 줄입니다. 이 작업은 클러스터 가동 중지 없이 수행됩니다.

```
aws elasticache decrease-replica-count \
    --replication-group-id my-cluster \
    --apply-immediately  \
    --new-replica-count 2
```

출력:

```
{
    "ReplicationGroup": {
        "ReplicationGroupId": "my-cluster",
        "Description": " ",
        "Status": "modifying",
        "PendingModifiedValues": {},
        "MemberClusters": [
            "myrepliace",
            "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": "myrepliace",
                        "CacheNodeId": "0001",
                        "ReadEndpoint": {
                            "Address":
 "myrepliace.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
    }
}
```

자세한 내용은 Elasticache 사용 설명서의 복제본 수 변경을 참조하세요.

- 자세한 API 내용은 명령 참조DecreaseReplicaCount의 섹션을 참조하세요. AWS CLI

### `delete-cache-cluster`

다음 코드 예시에서는 `delete-cache-cluster`을 사용하는 방법을 보여 줍니다.

AWS CLI

캐시 클러스터를 삭제하려면

다음 `delete-cache-cluster` 예제에서는 이전에 프로비저닝된 지정된 클러스터를 삭제합니다. 명령은 연결된 모든 캐시 노드, 노드 엔드포인트 및 클러스터 자체를 삭제합니다. 이 작업에서 성공적인 응답을 받으면 Amazon은 클러스터 삭제를 ElastiCache 즉시 시작합니다. 이 작업을 취소하거나 되돌릴 수 없습니다.

이 작업은 다음에는 유효하지 않습니다.

Redis(클러스터 모드 활성화됨) clustersA복제 groupA 마지막 읽기 전용 복제본인 클러스터다중 AZ 모드가 enabledA 노드 그룹(샤드)Redis(클러스터 모드 활성화됨) 복제 groupA 클러스터사용 가능한 상태가 아닌 클러스터

```
aws elasticache delete-cache-cluster \
    --cache-cluster-id "my-cluster-002"
```

출력:

```
{
    "CacheCluster": {
        "CacheClusterId": "my-cluster-002",
        "ClientDownloadLandingPage": "https://console.aws.amazon.com/elasticache/
home#client-download:",
        "CacheNodeType": "cache.r5.xlarge",
        "Engine": "redis",
        "EngineVersion": "5.0.5",
        "CacheClusterStatus": "deleting",
        "NumCacheNodes": 1,
        "PreferredAvailabilityZone": "us-west-2a",
        "CacheClusterCreateTime": "2019-11-26T03:35:04.546Z",
        "PreferredMaintenanceWindow": "mon:04:05-mon:05:05",
        "PendingModifiedValues": {},
        "NotificationConfiguration": {
            "TopicArn": "arn:aws:sns:us-west-x:xxxxxxx4152:My_Topic",
            "TopicStatus": "active"
        },
        "CacheSecurityGroups": [],
        "CacheParameterGroup": {
            "CacheParameterGroupName": "mygroup",
            "ParameterApplyStatus": "in-sync",
            "CacheNodeIdsToReboot": []
        },
        "CacheSubnetGroupName": "kxkxk",
        "AutoMinorVersionUpgrade": true,
        "SecurityGroups": [
            {
                "SecurityGroupId": "sg-xxxxxxxxxx9836",
                "Status": "active"
            },
            {
                "SecurityGroupId": "sg-xxxxxxxxxxxx7b",
                "Status": "active"
            }
        ],
        "ReplicationGroupId": "my-cluster",
        "SnapshotRetentionLimit": 0,
```

```
        "SnapshotWindow": "07:30-08:30",
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
}
```

자세한 내용은 Elasticache 사용 설명서의 클러스터 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteCacheCluster의 섹션을 참조하세요. AWS CLI

### delete-cache-parameter-group

다음 코드 예시에서는 delete-cache-parameter-group을 사용하는 방법을 보여 줍니다.

AWS CLI

캐시 파라미터 그룹을 삭제하려면

다음 delete-cache-parameter-group 예제에서는 지정된 캐시 파라미터 그룹을 삭제합니다. 캐시 클러스터와 연결된 캐시 파라미터 그룹은 삭제할 수 없습니다.

```
aws elasticache delete-cache-parameter-group \
    --cache-parameter-group-name myparamgroup
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Elasticache 사용 설명서의 파라미터 그룹 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteCacheParameterGroup의 섹션을 참조하세요. AWS CLI

### delete-cache-subnet-group

다음 코드 예시에서는 delete-cache-subnet-group을 사용하는 방법을 보여 줍니다.

AWS CLI

캐시 서브넷 그룹을 삭제하려면

다음 delete-cache-subnet-group 예제에서는 지정된 캐시 서브넷 그룹을 삭제합니다. 캐시 서브넷 그룹이 클러스터와 연결된 경우 삭제할 수 없습니다.

```
aws elasticache delete-cache-subnet-group \
    --cache-subnet-group-name "mygroup"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Elasticache 사용 설명서의 서브넷 그룹 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteCacheSubnetGroup의 섹션을 참조하세요. AWS CLI

### delete-global-replication-group

다음 코드 예시에서는 delete-global-replication-group을 사용하는 방법을 보여 줍니다.

AWS CLI

전역 복제 그룹을 삭제하려면

다음 delete-global-replication-group 예제에서는 새 전역 복제 그룹을 삭제합니다.

```
aws elasticache delete-global-replication-group \
    --global-replication-group-id my-global-replication-group \
    --retain-primary-replication-group
```

출력:

```
{
    "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
```

```
    }
}
```

자세한 내용은 Elasticache 사용 설명서의 글로벌 데이터 스토어를 사용하여 AWS 리전 간 복제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteGlobalReplicationGroup의 섹션을 참조하세요. AWS CLI

### `delete-replication-group`

다음 코드 예시에서는 `delete-replication-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

복제 그룹을 삭제하려면

다음 `delete-replication-group` 예제에서는 기존 복제 그룹을 삭제합니다. 기본적으로 이 작업은 기본/기본 및 모든 읽기 전용 복제본을 포함한 전체 복제 그룹을 삭제합니다. 복제 그룹에 기본 복제본이 하나만 있는 경우 RetainPrimaryCluster=true를 설정하여 기본 복제본을 유지하면서 선택적으로 읽기 전용 복제본만 삭제할 수 있습니다.

이 작업에서 성공적인 응답을 받으면 Amazon은 선택한 리소스 삭제를 ElastiCache 즉시 시작합니다. 이 작업을 취소하거나 되돌릴 수 없습니다. Redis에만 유효합니다.

```
aws elasticache delete-replication-group \
    --replication-group-id "mygroup"
```

출력:

```
{
     "ReplicationGroup": {
         "ReplicationGroupId": "mygroup",
         "Description": "my group",
         "Status": "deleting",
         "PendingModifiedValues": {},
         "AutomaticFailover": "disabled",
         "SnapshotRetentionLimit": 0,
         "SnapshotWindow": "06:00-07:00",
         "TransitEncryptionEnabled": false,
         "AtRestEncryptionEnabled": false
     }
 }
```

- 자세한 API 내용은 명령 참조DeleteReplicationGroup의 섹션을 참조하세요. AWS CLI

### delete-snapshot

다음 코드 예시에서는 delete-snapshot을 사용하는 방법을 보여 줍니다.

AWS CLI

스냅샷을 삭제하는 방법

다음 delete-snapshot 예제에서는 Redis 엔진을 사용하여 스냅샷을 삭제했습니다.

```
aws elasticache delete-snapshot \
    --snapshot-name mysnapshot
```

출력:

```
{
    "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:xxxxxxxxxxxx152: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"
            }
        ]
    }
}
```

자세한 내용은 Elasticache 사용 설명서의 Redis ElastiCache 용 백업 및 복원을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteSnapshot의 섹션을 참조하세요. AWS CLI

### `delete-user-group`

다음 코드 예시에서는 `delete-user-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 그룹을 삭제하려면

다음 delete-user-group 예제에서는 사용자 그룹을 삭제합니다.

```
aws elasticache delete-user-group \
    --user-group-id myusergroup
```

출력:

```
{
    "UserGroupId": "myusergroup",
    "Status": "deleting",
    "Engine": "redis",
    "UserIds": [
        "default"
    ],
    "ReplicationGroups": [],
    "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxx52:usergroup:myusergroup"
}
```

자세한 내용은 Elasticache 사용 설명서의 역할 기반 액세스 제어(RBAC)를 사용하여 사용자 인증을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteUserGroup의 섹션을 참조하세요. AWS CLI

### delete-user

다음 코드 예시에서는 delete-user을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 삭제

다음 delete-user 예제에서는 사용자를 삭제합니다.

```
aws elasticache delete-user \
    --user-id user2
```

출력:

```
{
    "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"
}
```

자세한 내용은 Elasticache 사용 설명서의 역할 기반 액세스 제어(RBAC)를 사용하여 사용자 인증을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteUser의 섹션을 참조하세요. AWS CLI

### `describe-cache-clusters`

다음 코드 예시에서는 `describe-cache-clusters`을 사용하는 방법을 보여 줍니다.

AWS CLI

캐시 클러스터를 설명하려면

다음 `describe-cache-clusters` 예제에서는 캐시 클러스터를 설명합니다.

```
aws elasticache describe-cache-clusters
```

출력:

```
{
    "CacheClusters": [
    {
            "CacheClusterId": "my-cluster-003",
            "ClientDownloadLandingPage": "https://console.aws.amazon.com/
elasticache/home#client-download:",
            "CacheNodeType": "cache.r5.large",
            "Engine": "redis",
            "EngineVersion": "5.0.5",
            "CacheClusterStatus": "available",
            "NumCacheNodes": 1,
            "PreferredAvailabilityZone": "us-west-2a",
```

```
            "CacheClusterCreateTime": "2019-11-26T01:22:52.396Z",
            "PreferredMaintenanceWindow": "mon:17:30-mon:18:30",
            "PendingModifiedValues": {},
            "NotificationConfiguration": {
                "TopicArn": "arn:aws:sns:us-west-2:xxxxxxxxxxx152:My_Topic",
                "TopicStatus": "active"
            },
            "CacheSecurityGroups": [],
            "CacheParameterGroup": {
                "CacheParameterGroupName": "default.redis5.0",
                "ParameterApplyStatus": "in-sync",
                "CacheNodeIdsToReboot": []
            },
            "CacheSubnetGroupName": "kxkxk",
            "AutoMinorVersionUpgrade": true,
            "SecurityGroups": [
                {
                    "SecurityGroupId": "sg-xxxxxd7b",
                    "Status": "active"
                }
            ],
            "ReplicationGroupId": "my-cluster",
            "SnapshotRetentionLimit": 0,
            "SnapshotWindow": "06:30-07:30",
            "AuthTokenEnabled": false,
            "TransitEncryptionEnabled": false,
            "AtRestEncryptionEnabled": false,
            "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxxx152:cluster:my-cache-
cluster",
            "ReplicationGroupLogDeliveryEnabled": false,
            "LogDeliveryConfigurations": [
                {
                    "LogType": "slow-log",
                    "DestinationType": "cloudwatch-logs",
                    "DestinationDetails": {
                        "CloudWatchLogsDetails": {
                            "LogGroup": "test-log"
                        }
                    },
                    "LogFormat": "text",
                    "Status": "active"
                }
            ]
        }
```

```
    ]
}
```

자세한 내용은 Elasticache 사용 설명서의 클러스터 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeCacheClusters의 섹션을 참조하세요. AWS CLI

### `describe-cache-engine-versions`

다음 코드 예시에서는 `describe-cache-engine-versions`을 사용하는 방법을 보여 줍니다.

AWS CLI

캐시 엔진 버전을 설명하려면

다음 `describe-cache-engine-versions` 예제에서는 사용 가능한 캐시 엔진 및 해당 버전의 목록을 반환합니다.

```
aws elasticache describe-cache-engine-versions \
    --engine "Redis"
```

출력:

```
{
    "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"
        }
    ]
```

```
}
```

- 자세한 API 내용은 명령 참조DescribeCacheEngineVersions의 섹션을 참조하세요. AWS CLI

### `describe-cache-parameter-groups`

다음 코드 예시에서는 `describe-cache-parameter-groups`을 사용하는 방법을 보여 줍니다.

AWS CLI

캐시 파라미터 그룹을 설명하려면

다음 `describe-cache-parameter-groups` 예제에서는 캐시 파라미터 그룹 설명 목록을 반환합니다.

```
aws elasticache describe-cache-parameter-groups \
    --cache-parameter-group-name "mygroup"
```

출력:

```
{
    "CacheParameterGroups": [
        {
            "CacheParameterGroupName": "mygroup",
            "CacheParameterGroupFamily": "redis5.0",
            "Description": " "
        }
    ]
}
```

자세한 내용은 Elasticache 사용 설명서의 파라미터 그룹을 사용하여 엔진 파라미터 구성을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeCacheParameterGroups의 섹션을 참조하세요. AWS CLI

### `describe-cache-parameters`

다음 코드 예시에서는 `describe-cache-parameters`을 사용하는 방법을 보여 줍니다.

AWS CLI

캐시 파라미터를 설명하려면

다음 "describe-cache-parameters" 예제는 지정된 캐시 파라미터 그룹에 대한 세부 파라미터 목록을 반환합니다.

```
aws elasticache describe-cache-parameters \
    --cache-parameter-group-name "myparamgroup"
```

출력:

```
{
    "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": "activerehashing",
            "ParameterValue": "yes",
            "Description": "Apply rehashing or not.",
            "Source": "user",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "requires-reboot"
        },
        {
            "ParameterName": "appendfsync",
            "ParameterValue": "everysec",
            "Description": "fsync policy for AOF persistence",
            "Source": "system",
            "DataType": "string",
            "AllowedValues": "always,everysec,no",
            "IsModifiable": false,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "appendonly",
            "ParameterValue": "no",
            "Description": "Enable Redis persistence.",
            "Source": "system",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "IsModifiable": false,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "client-output-buffer-limit-normal-hard-limit",
            "ParameterValue": "0",
            "Description": "Normal client output buffer hard limit in bytes.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
```

```
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "client-output-buffer-limit-normal-soft-limit",
            "ParameterValue": "0",
            "Description": "Normal client output buffer soft limit in bytes.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "client-output-buffer-limit-normal-soft-seconds",
            "ParameterValue": "0",
            "Description": "Normal client output buffer soft limit in seconds.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "client-output-buffer-limit-pubsub-hard-limit",
            "ParameterValue": "33554432",
            "Description": "Pubsub client output buffer hard limit in bytes.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "client-output-buffer-limit-pubsub-soft-limit",
            "ParameterValue": "8388608",
            "Description": "Pubsub client output buffer soft limit in bytes.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
```

```
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "client-output-buffer-limit-pubsub-soft-seconds",
            "ParameterValue": "60",
            "Description": "Pubsub client output buffer soft limit in seconds.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "client-output-buffer-limit-replica-soft-seconds",
            "ParameterValue": "60",
            "Description": "Replica client output buffer soft limit in seconds.",
            "Source": "system",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": false,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "client-query-buffer-limit",
            "ParameterValue": "1073741824",
            "Description": "Max size of a single client query buffer",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "1048576-1073741824",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "close-on-replica-write",
            "ParameterValue": "yes",
            "Description": "If enabled, clients who attempt to write to a read-only
replica will be disconnected. Applicable to 2.8.23 and higher.",
            "Source": "user",
            "DataType": "string",
```

```
            "AllowedValues": "yes,no",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "cluster-enabled",
            "ParameterValue": "no",
            "Description": "Enable cluster mode",
            "Source": "user",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "requires-reboot"
        },
        {
            "ParameterName": "cluster-require-full-coverage",
            "ParameterValue": "no",
            "Description": "Whether cluster becomes unavailable if one or more slots
are not covered",
            "Source": "user",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "databases",
            "ParameterValue": "16",
            "Description": "Set the number of databases.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "1-1200000",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "requires-reboot"
        },
        {
            "ParameterName": "hash-max-ziplist-entries",
            "ParameterValue": "512",
            "Description": "The maximum number of hash entries in order for the
dataset to be compressed.",
```

```
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "hash-max-ziplist-value",
            "ParameterValue": "64",
            "Description": "The threshold of biggest hash entries in order for the
dataset to be compressed.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "hll-sparse-max-bytes",
            "ParameterValue": "3000",
            "Description": "HyperLogLog sparse representation bytes limit",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "1-16000",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "lazyfree-lazy-eviction",
            "ParameterValue": "no",
            "Description": "Perform an asynchronous delete on evictions",
            "Source": "user",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "lazyfree-lazy-expire",
            "ParameterValue": "no",
```

```
            "Description": "Perform an asynchronous delete on expired keys",
            "Source": "user",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "lazyfree-lazy-server-del",
            "ParameterValue": "no",
            "Description": "Perform an asynchronous delete on key updates",
            "Source": "user",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "lfu-decay-time",
            "ParameterValue": "1",
            "Description": "The amount of time in minutes to decrement the key
counter for LFU eviction policy",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "lfu-log-factor",
            "ParameterValue": "10",
            "Description": "The log factor for incrementing key counter for LFU
eviction policy",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "1-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
```

```
            "ParameterName": "list-compress-depth",
            "ParameterValue": "0",
            "Description": "Number of quicklist ziplist nodes from each side of
the list to exclude from compression. The head and tail of the list are always
uncompressed for fast push/pop operations",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "list-max-ziplist-size",
            "ParameterValue": "-2",
            "Description": "The number of entries allowed per internal list node can
be specified as a fixed maximum size or a maximum number of elements",
            "Source": "system",
            "DataType": "integer",
            "AllowedValues": "-5,-4,-3,-2,-1,1-",
            "IsModifiable": false,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "lua-replicate-commands",
            "ParameterValue": "yes",
            "Description": "Always enable Lua effect replication or not",
            "Source": "user",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "lua-time-limit",
            "ParameterValue": "5000",
            "Description": "Max execution time of a Lua script in milliseconds. 0
for unlimited execution without warnings.",
            "Source": "system",
            "DataType": "integer",
            "AllowedValues": "5000",
            "IsModifiable": false,
```

```
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "maxclients",
            "ParameterValue": "65000",
            "Description": "The maximum number of Redis clients.",
            "Source": "system",
            "DataType": "integer",
            "AllowedValues": "1-65000",
            "IsModifiable": false,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "requires-reboot"
        },
        {
            "ParameterName": "maxmemory-policy",
            "ParameterValue": "volatile-lru",
            "Description": "Max memory policy.",
            "Source": "user",
            "DataType": "string",
            "AllowedValues": "volatile-lru,allkeys-lru,volatile-lfu,allkeys-
lfu,volatile-random,allkeys-random,volatile-ttl,noeviction",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "maxmemory-samples",
            "ParameterValue": "3",
            "Description": "Max memory samples.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "1-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "min-replicas-max-lag",
            "ParameterValue": "10",
            "Description": "The maximum amount of replica lag in seconds beyond
 which the master would stop taking writes. A value of 0 means the master always
 takes writes.",
            "Source": "user",
```

```
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "min-replicas-to-write",
            "ParameterValue": "0",
            "Description": "The minimum number of replicas that must be present with
lag no greater than min-replicas-max-lag for master to take writes. Setting this to
0 means the master always takes writes.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "notify-keyspace-events",
            "Description": "The keyspace events for Redis to notify Pub/Sub clients
about. By default all notifications are disabled",
            "Source": "user",
            "DataType": "string",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "proto-max-bulk-len",
            "ParameterValue": "536870912",
            "Description": "Max size of a single element request",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "1048576-536870912",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "rename-commands",
            "ParameterValue": "",
```

```
            "Description": "Redis commands that can be dynamically renamed by the
customer",
            "Source": "user",
            "DataType": "string",
            "AllowedValues":
"APPEND,BITCOUNT,BITFIELD,BITOP,BITPOS,BLPOP,BRPOP,BRPOPLPUSH,BZPOPMIN,BZPOPMAX,CLIENT,COMM
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.3",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "repl-backlog-size",
            "ParameterValue": "1048576",
            "Description": "The replication backlog size in bytes for PSYNC. This is
the size of the buffer which accumulates slave data when slave is disconnected for
some time, so that when slave reconnects again, only transfer the portion of data
which the slave missed. Minimum value is 16K.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "16384-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "repl-backlog-ttl",
            "ParameterValue": "3600",
            "Description": "The amount of time in seconds after the master no longer
have any slaves connected for the master to free the replication backlog. A value
of 0 means to never release the backlog.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "replica-allow-chaining",
            "ParameterValue": "no",
            "Description": "Configures if chaining of replicas is allowed",
            "Source": "system",
            "DataType": "string",
            "AllowedValues": "yes,no",
```

```
            "IsModifiable": false,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "replica-ignore-maxmemory",
            "ParameterValue": "yes",
            "Description": "Determines if replica ignores maxmemory setting by not
evicting items independent from the master",
            "Source": "system",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "IsModifiable": false,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "replica-lazy-flush",
            "ParameterValue": "no",
            "Description": "Perform an asynchronous flushDB during replica sync",
            "Source": "system",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "IsModifiable": false,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "reserved-memory-percent",
            "ParameterValue": "25",
            "Description": "The percent of memory reserved for non-cache memory
usage. You may want to increase this parameter for nodes with read replicas, AOF
enabled, etc, to reduce swap usage.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-100",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "set-max-intset-entries",
            "ParameterValue": "512",
```

```
            "Description": "The limit in the size of the set in order for the
dataset to be compressed.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "slowlog-log-slower-than",
            "ParameterValue": "10000",
            "Description": "The execution time, in microseconds, to exceed in order
for the command to get logged. Note that a negative number disables the slow log,
while a value of zero forces the logging of every command.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "slowlog-max-len",
            "ParameterValue": "128",
            "Description": "The length of the slow log. There is no limit to this
length. Just be aware that it will consume memory. You can reclaim memory used by
the slow log with SLOWLOG RESET.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "stream-node-max-bytes",
            "ParameterValue": "4096",
            "Description": "The maximum size of a single node in a stream in bytes",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
```

```
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "stream-node-max-entries",
            "ParameterValue": "100",
            "Description": "The maximum number of items a single node in a stream
can contain",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "tcp-keepalive",
            "ParameterValue": "300",
            "Description": "If non-zero, send ACKs every given number of seconds.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "timeout",
            "ParameterValue": "0",
            "Description": "Close connection if client is idle for a given number of
seconds, or never if 0.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0,20-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "zset-max-ziplist-entries",
            "ParameterValue": "128",
            "Description": "The maximum number of sorted set entries in order for
the dataset to be compressed.",
            "Source": "user",
            "DataType": "integer",
```

```
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        },
        {
            "ParameterName": "zset-max-ziplist-value",
            "ParameterValue": "64",
            "Description": "The threshold of biggest sorted set entries in order for
 the dataset to be compressed.",
            "Source": "user",
            "DataType": "integer",
            "AllowedValues": "0-",
            "IsModifiable": true,
            "MinimumEngineVersion": "5.0.0",
            "ChangeType": "immediate"
        }
    ]
}
```

자세한 내용은 Elasticache 사용 설명서의 파라미터 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeCacheParameters의 섹션을 참조하세요. AWS CLI

### describe-cache-subnet-groups

다음 코드 예시에서는 describe-cache-subnet-groups을 사용하는 방법을 보여 줍니다.

AWS CLI

캐시 서브넷 그룹을 설명하려면

다음 describe-cache-subnet-groups 예제에서는 서브넷 그룹 목록을 반환합니다.

```
aws elasticache describe-cache-subnet-groups
```

출력:

```
{
    "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"
                    }
                }
            ]
        }
    ]
}
```

자세한 내용은 Elasticache 사용 설명서의 서브넷 및 서브넷 그룹 또는 ElastiCache Memcached 사용 설명서의 서브넷 및 서브넷 그룹을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeCacheSubnetGroups의 섹션을 참조하세요. AWS CLI

### `describe-engine-default-parameters`

다음 코드 예시에서는 `describe-engine-default-parameters`을 사용하는 방법을 보여 줍니다.

AWS CLI

엔진 기본 파라미터를 설명하려면

다음 `describe-engine-default-parameters` 예제에서는 지정된 캐시 엔진에 대한 기본 엔진 및 시스템 파라미터 정보를 반환합니다.

```
aws elasticache describe-engine-default-parameters \
    --cache-parameter-group-family "redis5.0"
```

출력:

```
{
    "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": "activerehashing",
                "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"
            }
        ]
    }
}
```

- 자세한 API 내용은 명령 참조DescribeEngineDefaultParameters의 섹션을 참조하세요. AWS CLI

### `describe-events`

다음 코드 예시에서는 `describe-events`을 사용하는 방법을 보여 줍니다.

AWS CLI

복제 그룹의 이벤트를 설명하려면

다음 `describe-events` 예제에서는 복제 그룹에 대한 이벤트 목록을 반환합니다.

```
aws elasticache describe-events \
    --source-identifier test-cluster \
    --source-type replication-group
```

출력:

```
{
    "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"
        }
    ]
```

```
}
```

자세한 내용은 Elasticache 사용 설명서의 이벤트 모니터링을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeEvents의 섹션을 참조하세요. AWS CLI

### describe-global-replication-groups

다음 코드 예시에서는 describe-global-replication-groups을 사용하는 방법을 보여 줍니다.

AWS CLI

글로벌 복제 그룹을 설명하려면

다음 describe-global-replication-groups 예제에서는 글로벌 데이터 스토어의 세부 정보를 반환합니다.

```
aws elasticache describe-global-replication-groups \
    --global-replication-group-id my-grg
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 Elasticache 사용 설명서의 글로벌 데이터 스토어를 사용하여 AWS 리전 간 복제를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeGlobalReplicationGroups의 섹션을 참조하세요. AWS CLI

### describe-replication-groups

다음 코드 예시에서는 describe-replication-groups을 사용하는 방법을 보여 줍니다.

AWS CLI

복제 그룹 세부 정보 목록을 반환하려면

다음 describe-replication-groups 예제에서는 복제 그룹을 반환합니다.

```
aws elasticache describe-replication-groups
```

출력:

```
{
    "ReplicationGroups": [
        {
            "ReplicationGroupId": "my-cluster",
            "Description": "mycluster",
            "Status": "available",
            "PendingModifiedValues": {},
            "MemberClusters": [
                "pat-cluster-001",
                "pat-cluster-002",
                "pat-cluster-003",
                "pat-cluster-004"
            ],
            "NodeGroups": [
                {
                    "NodeGroupId": "0001",
                    "Status": "available",
                    "PrimaryEndpoint": {
                        "Address": "my-
cluster.xxxxih.ng.0001.usw2.cache.amazonaws.com",
                        "Port": 6379
                    },
                    "ReaderEndpoint": {
                        "Address": "my-cluster-
ro.xxxxih.ng.0001.usw2.cache.amazonaws.com",
                        "Port": 6379
                    },
                    "NodeGroupMembers": [
                        {
                            "CacheClusterId": "my-cluster-001",
```

```
                            "CacheNodeId": "0001",
                            "ReadEndpoint": {
                                "Address": "pat-
cluster-001.xxxih.0001.usw2.cache.amazonaws.com",
                                "Port": 6379
                            },
                            "PreferredAvailabilityZone": "us-west-2a",
                            "CurrentRole": "primary"
                        },
                        {
                            "CacheClusterId": "my-cluster-002",
                            "CacheNodeId": "0001",
                            "ReadEndpoint": {
                                "Address": "pat-
cluster-002.xxxxih.0001.usw2.cache.amazonaws.com",
                                "Port": 6379
                            },
                            "PreferredAvailabilityZone": "us-west-2a",
                            "CurrentRole": "replica"
                        },
                        {
                            "CacheClusterId": "my-cluster-003",
                            "CacheNodeId": "0001",
                            "ReadEndpoint": {
                                "Address": "pat-
cluster-003.xxxxih.0001.usw2.cache.amazonaws.com",
                                "Port": 6379
                            },
                            "PreferredAvailabilityZone": "us-west-2a",
                            "CurrentRole": "replica"
                        },
                        {
                            "CacheClusterId": "my-cluster-004",
                            "CacheNodeId": "0001",
                            "ReadEndpoint": {
                                "Address": "pat-
cluster-004.xxxih.0001.usw2.cache.amazonaws.com",
                                "Port": 6379
                            },
                            "PreferredAvailabilityZone": "us-west-2a",
                            "CurrentRole": "replica"
                        }
                    ]
                }
```

```
            ],
            "AutomaticFailover": "disabled",
            "SnapshotRetentionLimit": 0,
            "SnapshotWindow": "07:30-08:30",
            "ClusterEnabled": false,
            "CacheNodeType": "cache.r5.xlarge",
            "AuthTokenEnabled": false,
            "TransitEncryptionEnabled": false,
            "AtRestEncryptionEnabled": false,
            "ARN": "arn:aws:elasticache:us-
west-2:xxxxxxxxxxx152:replicationgroup:my-cluster",
            "LogDeliveryConfigurations": [
                {
                    "LogType": "slow-log",
                    "DestinationType": "cloudwatch-logs",
                    "DestinationDetails": {
                        "CloudWatchLogsDetails": {
                            "LogGroup": "test-log"
                        }
                    },
                    "LogFormat": "json",
                    "Status": "active"
                }
            ]
        }
    ]
}
```

자세한 내용은 Elasticache 사용 설명서의 클러스터 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeReplicationGroups의 섹션을 참조하세요. AWS CLI

### `describe-reserved-cache-nodes-offerings`

다음 코드 예시에서는 `describe-reserved-cache-nodes-offerings`을 사용하는 방법을 보여 줍니다.

AWS CLI

설명하려면 reserved-cache-nodes-offerings

다음 `describe-reserved-cache-nodes-offerings` 예제에서는 옵션의 세부 정보를 반환합니다 reserved-cache-node.

```
aws elasticache describe-reserved-cache-nodes-offerings
```

출력:

```
{
    "ReservedCacheNodesOfferings": [
        {
            "ReservedCacheNodesOfferingId": "01ce0a19-a476-41cb-8aee-48eacbcdc8e5",
            "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"
                }
            ]
        },

        ...

    ]
}
```

자세한 내용은 Elasticache Redis 사용 설명서의 예약 노드 제안에 대한 정보 가져오기 또는 Elasticache Memcached 사용 설명서의 예약 노드 제안에 대한 정보 가져오기를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeReservedCacheNodesOfferings의 섹션을 참조하세요. AWS CLI

### `describe-reserved-cache-nodes`

다음 코드 예시에서는 `describe-reserved-cache-nodes`을 사용하는 방법을 보여 줍니다.

AWS CLI

예약 캐시 노드를 설명하려면

다음 `describe-reserved-cache-nodes` 예제에서는 이 계정에 대해 예약된 캐시 노드 또는 지정된 예약 캐시 노드에 대한 정보를 반환합니다.

aws elasticache describe-reserved-cache-nodes

출력:

```
{
    "ReservedCacheNodes": [
        {
            "ReservedCacheNodeId": "mynode",
            "ReservedCacheNodesOfferingId": "xxxxxxxxx-xxxxx-xxxxx-xxxx-xxxxxxxx71",
            "CacheNodeType": "cache.t3.small",
            "StartTime": "2019-12-06T02:50:44.003Z",
            "Duration": 31536000,
            "FixedPrice": 0.0,
            "UsagePrice": 0.0,
            "CacheNodeCount": 1,
            "ProductDescription": "redis",
            "OfferingType": "No Upfront",
            "State": "payment-pending",
            "RecurringCharges": [
                {
                    "RecurringChargeAmount": 0.023,
                    "RecurringChargeFrequency": "Hourly"
                }
            ],
            "ReservationARN": "arn:aws:elasticache:us-
west-2:xxxxxxxxxxx52:reserved-instance:mynode"
        }
```

```
    ]
}
```

자세한 내용은 Elasticache 사용 설명서의 예약 노드로 비용 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeReservedCacheNodes의 섹션을 참조하세요. AWS CLI

### describe-service-updates

다음 코드 예시에서는 describe-service-updates을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스 업데이트를 설명하려면

다음 describe-service-updates 예제에서는 서비스 업데이트에 대한 세부 정보를 반환합니다.

```
aws elasticache describe-service-updates
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeServiceUpdates의 섹션을 참조하세요. AWS CLI

### `describe-snapshots`

다음 코드 예시에서는 `describe-snapshots`을 사용하는 방법을 보여 줍니다.

AWS CLI

스냅샷을 설명하려면

다음 'describe-snapshots' 예제는 클러스터 또는 복제 그룹 스냅샷에 대한 정보를 반환합니다.

```
aws elasticache describe-snapshots
```

출력:

```
{
    "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"
                }
            ]
        }
    ]
}
```

자세한 내용은 Elasticache 사용 설명서의 Redis ElastiCache 용 백업 및 복원을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeSnapshots의 섹션을 참조하세요. AWS CLI

### `describe-update-actions`

다음 코드 예시에서는 `describe-update-actions`을 사용하는 방법을 보여 줍니다.

AWS CLI

업데이트 작업을 설명하려면

다음 `describe-update-actions` 예제에서는 업데이트 작업에 대한 세부 정보를 반환합니다.

```
aws elasticache describe-update-actions
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Elasticache 사용 설명서의 Amazon에서 셀프 서비스 업데이트를 참조하세요 ElastiCache.

- 자세한 API 내용은 명령 참조DescribeUpdateActions의 섹션을 참조하세요. AWS CLI

### `describe-user-groups`

다음 코드 예시에서는 `describe-user-groups`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 그룹을 설명하려면

다음 describe-user-groups 예제에서는 사용자 그룹 목록을 반환합니다.

```
aws elasticache describe-user-groups
```

출력:

```
{
    "UserGroups": [
        {
            "UserGroupId": "myusergroup",
            "Status": "active",
            "Engine": "redis",
            "UserIds": [
                "default"
            ],
            "ReplicationGroups": [],
            "ARN": "arn:aws:elasticache:us-
west-2:xxxxxxxxxx52:usergroup:myusergroup"
        }
    ]
}
```

자세한 내용은 Elasticache 사용 설명서의 역할 기반 액세스 제어(RBAC)를 사용하여 사용자 인증을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeUserGroups의 섹션을 참조하세요. AWS CLI

### describe-users

다음 코드 예시에서는 describe-users을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자를 설명하려면

다음 describe-users 예제에서는 사용자 목록을 반환합니다.

```
aws elasticache describe-users
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Elasticache 사용 설명서의 역할 기반 액세스 제어(RBAC)를 사용하여 사용자 인증을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeUsers의 섹션을 참조하세요. AWS CLI

### disassociate-global-replication-group

다음 코드 예시에서는 disassociate-global-replication-group을 사용하는 방법을 보여 줍니다.

AWS CLI

글로벌 복제 그룹에서 보조 클러스터를 연결하려면

다음 disassociate-global-replication-group 예제에서는 글로벌 데이터 스토어에서 보조 클러스터를 제거합니다.

```
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
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 Elasticache 사용 설명서의 글로벌 데이터 스토어를 사용하여 AWS 리전 간 복제를 참조하세요.

- 자세한 API 내용은 명령 참조DisassociateGlobalReplicationGroup의 섹션을 참조하세요. AWS CLI

### increase-node-groups-in-global-replication-group

다음 코드 예시에서는 increase-node-groups-in-global-replication-group을 사용하는 방법을 보여 줍니다.

AWS CLI

전역 복제 그룹의 노드 그룹 수를 늘리려면

다음은 Redis 엔진을 사용하여 노드 그룹 수를 increase-node-groups-in-global-replication-group 늘립니다.

```
aws elasticache increase-node-groups-in-global-replication-group \
    --global-replication-group-id sgaui-pat-test-4 \
    --node-group-count 6 \
    --apply-immediately
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 Elasticache 사용 설명서의 글로벌 데이터 스토어를 사용하여 AWS 리전 간 복제를 참조하세요.

- 자세한 API 내용은 명령 참조IncreaseNodeGroupsInGlobalReplicationGroup의 섹션을 참조하세요. AWS CLI

### `increase-replica-count`

다음 코드 예시에서는 `increase-replica-count`을 사용하는 방법을 보여 줍니다.

AWS CLI

복제본 수를 늘리려면

다음 `increase-replica-count` 예제에서는 두 가지 중 하나를 수행합니다. Redis(클러스터 모드 비활성화됨) 복제 그룹의 복제본 수를 동적으로 늘릴 수 있습니다. 또는 Redis(클러스터 모드 활성화됨) 복제 그룹의 하나 이상의 노드 그룹(샤드)에서 복제본 노드 수를 동적으로 늘릴 수 있습니다. 이 작업은 클러스터 가동 중지 없이 수행됩니다.

```
aws elasticache increase-replica-count \
    --replication-group-id "my-cluster" \
    --apply-immediately \
    --new-replica-count 3
```

출력:

```
{
     "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
    }
}
```

자세한 내용은 Elasticache 사용 설명서의 샤드에서 복제본 수 증가를 참조하세요.

- 자세한 API 내용은 명령 참조IncreaseReplicaCount의 섹션을 참조하세요. AWS CLI

### `list-allowed-node-type-modifications`

다음 코드 예시에서는 `list-allowed-node-type-modifications`을 사용하는 방법을 보여 줍니다.

AWS CLI

허용된 노드 수정을 나열하려면

다음 `list-allowed-node-type-modifications` 예제에서는 Redis 클러스터 또는 복제 그룹의 현재 노드 유형을 확장할 수 있는 사용 가능한 모든 노드 유형을 나열합니다.

```
aws elasticache list-allowed-node-type-modifications \
    --replication-group-id "my-replication-group"
```

출력:

```
{
    "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"
    ]
}
```

자세한 내용은 Elasticache 사용 설명서의 Redis 클러스터에 ElastiCache 대한 확장을 참조하세요.

- 자세한 API 내용은 명령 참조ListAllowedNodeTypeModifications의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스의 태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 리소스에 대한 태그를 나열합니다.

```
aws elasticache list-tags-for-resource \
    --resource-name "arn:aws:elasticache:us-east-1:123456789012:cluster:my-cluster"
```

출력:

```
{
```

```
    "TagList": [
        {
            "Key": "Project",
            "Value": "querySpeedUp"
        },
        {
            "Key": "Environment",
            "Value": "PROD"
        }
    ]
}
```

자세한 내용은 Elasticache 사용 설명서의 를 사용하여 태그 나열 AWS CLI을 참조하세요.

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### `modify-cache-cluster`

다음 코드 예시에서는 `modify-cache-cluster`을 사용하는 방법을 보여 줍니다.

AWS CLI

캐시 클러스터를 수정하려면

다음 `modify-cache-cluster` 예제에서는 지정된 클러스터에 대한 설정을 수정합니다.

```
aws elasticache modify-cache-cluster \
    --cache-cluster-id "my-cluster" \
    --num-cache-nodes 1
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 Elasticache 사용 설명서의 ElastiCache 클러스터 수정을 참조하세요.

- 자세한 API 내용은 명령 참조ModifyCacheCluster의 섹션을 참조하세요. AWS CLI

### `modify-cache-parameter-group`

다음 코드 예시에서는 `modify-cache-parameter-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

캐시 파라미터 그룹을 수정하려면

다음 `modify-cache-parameter-group` 예제에서는 지정된 캐시 파라미터 그룹의 파라미터를 수정합니다.

```
aws elasticache modify-cache-parameter-group \
    --cache-parameter-group-name "mygroup" \
    --parameter-name-values "ParameterName=activedefrag, ParameterValue=no"
```

출력:

```
{
    "CacheParameterGroupName": "mygroup"
}
```

자세한 내용은 Elasticache 사용 설명서의 파라미터 그룹 수정을 참조하세요.

- 자세한 API 내용은 명령 참조ModifyCacheParameterGroup의 섹션을 참조하세요. AWS CLI

### modify-cache-subnet-group

다음 코드 예시에서는 modify-cache-subnet-group을 사용하는 방법을 보여 줍니다.

AWS CLI

캐시 서브넷 그룹을 수정하려면

다음 modify-cache-subnet-group 예제에서는 지정된 캐시 서브넷 그룹을 수정합니다.

```
aws elasticache modify-cache-subnet-group \
    --cache-subnet-group-name kxkxk \
    --cache-subnet-group-description "mygroup"
```

출력:

```
{
    "CacheSubnetGroup": {
        "CacheSubnetGroupName": "kxkxk",
        "CacheSubnetGroupDescription": "mygroup",
        "VpcId": "vpc-xxxxcdb",
        "Subnets": [
            {
                "SubnetIdentifier": "subnet-xxxxbff",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2a"
                }
            }
        ]
    }
}
```

자세한 내용은 Elasticache 사용 설명서의 서브넷 그룹 수정을 참조하세요.

- 자세한 API 내용은 명령 참조ModifyCacheSubnetGroup의 섹션을 참조하세요. AWS CLI

### modify-global-replication-group

다음 코드 예시에서는 modify-global-replication-group을 사용하는 방법을 보여 줍니다.

AWS CLI

전역 복제 그룹을 수정하려면

다음은 Redis 엔진을 사용하여 자동 장애 조치를 비활성화하는 글로벌 복제 그룹의 속성을 modify-global-replication-group 수정합니다.

```
aws elasticache modify-global-replication-group \
    --global-replication-group-id sgaui-pat-group \
    --apply-immediately \
    --no-automatic-failover-enabled
```

출력

```
{
    "GlobalReplicationGroup": {
        "GlobalReplicationGroupId": "sgaui-test-group",
        "GlobalReplicationGroupDescription": " ",
        "Status": "modifying",
        "CacheNodeType": "cache.r5.large",
        "Engine": "redis",
        "EngineVersion": "5.0.6",
        "ClusterEnabled": false,
        "AuthTokenEnabled": false,
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
}
```

자세한 내용은 Elasticache 사용 설명서의 글로벌 데이터 스토어를 사용하여 AWS 리전 간 복제를 참조하세요.

- 자세한 API 내용은 명령 참조ModifyGlobalReplicationGroup의 섹션을 참조하세요. AWS CLI

### modify-replication-group-shard-configuration

다음 코드 예시에서는 modify-replication-group-shard-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

복제 그룹 샤드 구성을 수정하려면

다음은 Redis 엔진을 사용하여 노드 그룹 수를 modify-replication-group-shard-configuration 줄입니다.

```
aws elasticache modify-replication-group-shard-configuration \
    --replication-group-id mycluster \
    --node-group-count 3 \
    --apply-immediately \
    --node-groups-to-remove 0002
```

출력

```
{
    "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
    }
}
```

자세한 내용은 Elasticache 사용 설명서의 Redis 클러스터에 ElastiCache 대한 크기 조정을 참조하세요.

- 자세한 API 내용은 명령 참조ModifyReplicationGroupShardConfiguration의 섹션을 참조하세요. AWS CLI

### modify-replication-group

다음 코드 예시에서는 modify-replication-group을 사용하는 방법을 보여 줍니다.

AWS CLI

복제 그룹을 수정하려면

다음은 Redis 엔진을 사용하여 다중 AZ를 modify-replication-group 비활성화합니다.

```
aws elasticache modify-replication-group \
    --replication-group-id test-cluster \
    --no-multi-az-enabled \
    --apply-immediately
```

출력

```
{
    "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
    }
}
```

자세한 내용은 Elasticache 사용 설명서의 복제 그룹 수정을 참조하세요.

- 자세한 API 내용은 명령 참조ModifyReplicationGroup의 섹션을 참조하세요. AWS CLI

### `modify-user-group`

다음 코드 예시에서는 `modify-user-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 그룹을 수정하려면

다음 `modify-user-group` 예제에서는 사용자를 사용자 그룹에 추가합니다.

```
aws elasticache modify-user-group \
    --user-group-id myusergroup \
```

```
    --user-ids-to-add user1
```

출력:

```
{
    "UserGroupId": "myusergroup",
    "Status": "modifying",
    "Engine": "redis",
    "UserIds": [
        "default"
    ],
    "PendingChanges": {
        "UserIdsToAdd": [
            "user1"
        ]
    },
    "ReplicationGroups": [],
    "ARN": "arn:aws:elasticache:us-west-2:xxxxxxxxxx52:usergroup:myusergroup"
}
```

자세한 내용은 Elasticache 사용 설명서의 역할 기반 액세스 제어를 사용하여 사용자 인증(RBAC)을 참조하세요.

- 자세한 API 내용은 명령 참조ModifyUserGroup의 섹션을 참조하세요. AWS CLI

### modify-user

다음 코드 예시에서는 modify-user을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자를 수정하려면

다음 modify-user 예제에서는 사용자의 액세스 문자열을 수정합니다.

```
aws elasticache modify-user \
    --user-id user2 \
    --append-access-string "on ~* +@all"
```

출력:

```
{
```

```
    "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"
}
```

자세한 내용은 Elasticache 사용 설명서의 역할 기반 액세스 제어(RBAC)를 사용하여 사용자 인증을 참조하세요.

- 자세한 API 내용은 명령 참조ModifyUser의 섹션을 참조하세요. AWS CLI

### `purchase-reserved-cache-nodes-offering`

다음 코드 예시에서는 `purchase-reserved-cache-nodes-offering`을 사용하는 방법을 보여 줍니다.

AWS CLI

를 구매하려면 reserved-cache-node-offering

다음 `purchase-reserved-cache-nodes-offering` 예제에서는 예약 캐시 노드 제품을 구매할 수 있습니다.

```
aws elasticache purchase-reserved-cache-nodes-offering \
    --reserved-cache-nodes-offering-id xxxxxxx-4da5-4b90-b92d-929fbd7abed2
```

출력

```
{
    "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"
            }
        ]
    }
}
```

자세한 내용은 Elasticache Redis 사용 설명서의 예약 노드 제안에 대한 정보 가져오기 또는 Elasticache Memcached 사용 설명서의 예약 노드 제안에 대한 정보 가져오기를 참조하세요.

- 자세한 API 내용은 명령 참조PurchaseReservedCacheNodesOffering의 섹션을 참조하세요. AWS CLI

### `reboot-cache-cluster`

다음 코드 예시에서는 `reboot-cache-cluster`을 사용하는 방법을 보여 줍니다.

AWS CLI

캐시 클러스터를 재부팅하려면

다음 `reboot-cache-cluster` 예제에서는 프로비저닝된 클러스터 내에서 캐시 노드의 일부 또는 전부를 재부팅합니다. 이 작업은 수정된 캐시 파라미터 그룹을 클러스터에 적용합니다. 재부팅 작업은 가능한 한 빨리 수행되며 클러스터가 일시적으로 중단됩니다. 재부팅 중에 클러스터 상태는 로 설정됩니다`REBOOTING`.

```
aws elasticache reboot-cache-cluster \
    --cache-cluster-id "my-cluster-001" \
    --cache-node-ids-to-reboot "0001"
```

출력:

```
{
    "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-xxxxxxxxxxxxx836",
                "Status": "active"
            },
            {
                "SecurityGroupId": "sg-xxxxxxxx7b",
                "Status": "active"
            }
        ],
        "ReplicationGroupId": "my-cluster",
        "SnapshotRetentionLimit": 0,
        "SnapshotWindow": "07:30-08:30",
        "TransitEncryptionEnabled": false,
        "AtRestEncryptionEnabled": false
    }
}
```

자세한 내용은 Elasticache 사용 설명서의 클러스터 재부팅 <https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/Clusters.Rebooting.html을 참조하세요.

- 자세한 API 내용은 명령 참조RebootCacheCluster의 섹션을 참조하세요. AWS CLI

### `reset-cache-parameter-group`

다음 코드 예시에서는 `reset-cache-parameter-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

캐시 파라미터 그룹을 재설정하려면

다음 `reset-cache-parameter-group` 예제에서는 캐시 파라미터 그룹의 파라미터를 엔진 또는 시스템 기본값으로 수정합니다. 파라미터 이름 목록을 제출하여 특정 파라미터를 재설정할 수 있습니다. 전체 캐시 파라미터 그룹을 재설정하려면 `--reset-all-parameters` 및 `--cache-parameter-group-name` 파라미터를 지정합니다.

```
aws elasticache reset-cache-parameter-group \
    --cache-parameter-group-name "mygroup" \
    --reset-all-parameters
```

출력:

```
{
    "CacheParameterGroupName": "mygroup"
}
```

- 자세한 API 내용은 명령 참조ResetCacheParameterGroup의 섹션을 참조하세요. AWS CLI

### `start-migration`

다음 코드 예시에서는 `start-migration`을 사용하는 방법을 보여 줍니다.

AWS CLI

마이그레이션을 시작하려면

다음은 Redis 엔진을 ElastiCache사용하여 데이터를 Amazon의 자체 호스팅 Redis에서 Amazon EC2로 `start-migration` 마이그레이션합니다.

```
aws elasticache start-migration \
   --replication-group-id test \
```

```
   --customer-node-endpoint-
list "Address='test.g2xbih.ng.0001.usw2.cache.amazonaws.com',Port=6379"
```

출력

```
{
    "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
    }
}
```

자세한 내용은 Elasticache 사용 설명서의 로 온라인 마이그레이션 ElastiCache을 참조하세요.

- 자세한 API 내용은 명령 참조StartMigration의 섹션을 참조하세요. AWS CLI

### `test-failover`

다음 코드 예시에서는 `test-failover`을 사용하는 방법을 보여 줍니다.

AWS CLI

노드 그룹의 장애 조치를 테스트하려면

다음 `test-failover` 예제에서는 복제 그룹(콘솔에서 클러스터라고 함)의 지정된 노드 그룹(콘솔에서 샤드라고 함)에서 자동 장애 조치를 테스트합니다.

```
aws elasticache test-failover /
    --replication-group-id "mycluster" /
    --node-group-id "0001"
```

출력:

```
{
    "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
    }
}
```

- 자세한 API 내용은 명령 참조TestFailover의 섹션을 참조하세요. AWS CLI

## MediaStore 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 MediaStore.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### `create-container`

다음 코드 예시에서는 `create-container`을 사용하는 방법을 보여 줍니다.

AWS CLI

컨테이너를 생성하려면

다음 `create-container` 예제에서는 빈 새 컨테이너를 생성합니다.

```
aws mediastore create-container --container-name ExampleContainer
```

출력:

```
{
    "Container": {
        "AccessLoggingEnabled": false,
        "CreationTime": 1563557265,
        "Name": "ExampleContainer",
        "Status": "CREATING",
        "ARN": "arn:aws:mediastore:us-west-2:111122223333:container/
ExampleContainer"
    }
}
```

자세한 내용은 요소 사용 설명서의 컨테이너 생성을 참조하세요. AWS MediaStore

- 자세한 API 내용은 명령 참조CreateContainer의 섹션을 참조하세요. AWS CLI

### `delete-container-policy`

다음 코드 예시에서는 `delete-container-policy`을 사용하는 방법을 보여 줍니다.

AWS CLI

컨테이너 정책을 삭제하려면

다음 `delete-container-policy` 예제에서는 지정된 컨테이너에 할당된 정책을 삭제합니다. 정책이 삭제되면 AWS Elemental은 컨테이너에 기본 정책을 MediaStore 자동으로 할당합니다.

```
aws mediastore delete-container-policy \
    --container-name LiveEvents
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 요소 참조DeleteContainerPolicy의 섹션을 참조하세요. AWS MediaStore API

- 자세한 API 내용은 명령 참조DeleteContainerPolicy의 섹션을 참조하세요. AWS CLI

### delete-container

다음 코드 예시에서는 delete-container을 사용하는 방법을 보여 줍니다.

AWS CLI

컨테이너를 삭제하려면

다음 delete-container 예제에서는 지정된 컨테이너를 삭제합니다. 객체가 들어 있지 않은 컨테이너만 삭제할 수 있습니다.

```
aws mediastore delete-container \
    --container-name=ExampleLiveDemo
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS 요소 MediaStore 사용 설명서의 컨테이너 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteContainer의 섹션을 참조하세요. AWS CLI

### delete-cors-policy

다음 코드 예시에서는 delete-cors-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

CORS 정책을 삭제하려면

다음 delete-cors-policy 예제에서는 지정된 컨테이너에 할당된 크로스 오리진 리소스 공유(CORS) 정책을 삭제합니다.

```
aws mediastore delete-cors-policy \
```

```
    --container-name ExampleContainer
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS 요소 MediaStore 사용 설명서의 CORS 정책 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteCorsPolicy의 섹션을 참조하세요. AWS CLI

### delete-lifecycle-policy

다음 코드 예시에서는 delete-lifecycle-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

객체 수명 주기 정책을 삭제하려면

다음 delete-lifecycle-policy 예제에서는 지정된 컨테이너에 연결된 객체 수명 주기 정책을 삭제합니다. 이 변경 사항을 적용하려면 최대 20분이 걸릴 수 있습니다.

```
aws mediastore delete-lifecycle-policy \
    --container-name LiveEvents
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS 요소 MediaStore 사용 설명서의 객체 수명 주기 정책 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteLifecyclePolicy의 섹션을 참조하세요. AWS CLI

### describe-container

다음 코드 예시에서는 describe-container을 사용하는 방법을 보여 줍니다.

AWS CLI

컨테이너의 세부 정보를 보려면

다음 describe-container 예제에서는 지정된 컨테이너의 세부 정보를 표시합니다.

```
aws mediastore describe-container \
    --container-name ExampleContainer
```

출력:

```
{
    "Container": {
        "CreationTime": 1563558086,
        "AccessLoggingEnabled": false,
        "ARN": "arn:aws:mediastore:us-west-2:111122223333:container/
ExampleContainer",
        "Status": "ACTIVE",
        "Name": "ExampleContainer",
        "Endpoint": "https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com"
    }
}
```

자세한 내용은 AWS 요소 MediaStore 사용 설명서의 컨테이너 세부 정보 보기를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeContainer의 섹션을 참조하세요. AWS CLI

### describe-object

다음 코드 예시에서는 describe-object을 사용하는 방법을 보여 줍니다.

AWS CLI

특정 컨테이너에서 객체 및 폴더 목록을 보려면

다음 describe-object 예제에서는 특정 컨테이너에 저장된 항목(객체 및 폴더)을 보여줍니다.

```
aws mediastore-data describe-object \
    --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com \
    --path /folder_name/file1234.jpg
```

출력:

```
{
    "ContentType": "image/jpeg",
    "LastModified": "Fri, 19 Jul 2019 21:32:20 GMT",
    "ContentLength": "2307346",
    "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3"
}
```

자세한 내용은 AWS 요소 MediaStore 사용 설명서의 객체 세부 정보 보기를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeObject의 섹션을 참조하세요. AWS CLI

### get-container-policy

다음 코드 예시에서는 get-container-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

컨테이너 정책을 보려면

다음 get-container-policy 예제에서는 지정된 컨테이너의 리소스 기반 정책을 보여줍니다.

```
aws mediastore get-container-policy \
    --container-name ExampleLiveDemo
```

출력:

```
{
    "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"
                    }
                }
            }
        ]
    }
}
```

자세한 내용은 요소 사용 설명서의 컨테이너 정책 보기를 참조하세요. AWS MediaStore

- 자세한 API 내용은 명령 참조GetContainerPolicy의 섹션을 참조하세요. AWS CLI

### get-cors-policy

다음 코드 예시에서는 get-cors-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

CORS 정책을 보려면

다음 get-cors-policy 예제에서는 지정된 컨테이너에 할당된 크로스 오리진 리소스 공유(CORS) 정책을 보여줍니다.

```
aws mediastore get-cors-policy \
    --container-name ExampleContainer \
    --region us-west-2
```

출력:

```
{
    "CorsPolicy": [
        {
            "AllowedMethods": [
                "GET",
                "HEAD"
            ],
            "MaxAgeSeconds": 3000,
            "AllowedOrigins": [
                ""
            ],
            "AllowedHeaders": [
                ""
            ]
        }
    ]
}
```

자세한 내용은 AWS 요소 MediaStore 사용 설명서의 CORS 정책 보기를 참조하세요.

- 자세한 API 내용은 명령 참조GetCorsPolicy의 섹션을 참조하세요. AWS CLI

### get-lifecycle-policy

다음 코드 예시에서는 get-lifecycle-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

객체 수명 주기 정책을 보려면

다음 `get-lifecycle-policy` 예제에서는 지정된 컨테이너에 연결된 객체 수명 주기 정책을 보여줍니다.

```
aws mediastore get-lifecycle-policy \
    --container-name LiveEvents
```

출력:

```
{
    "LifecyclePolicy": {
        "rules": [
            {
                "definition": {
                    "path": [
                        {
                            "prefix": "Football/"
                        },
                        {
                            "prefix": "Baseball/"
                        }
                    ],
                    "days_since_create": [
                        {
                            "numeric": [
                                ">",
                                28
                            ]
                        }
                    ]
                },
                "action": "EXPIRE"
            }
        ]
    }
}
```

자세한 내용은 요소 사용 설명서의 객체 수명 주기 정책 보기를 참조하세요. AWS MediaStore

- 자세한 API 내용은 명령 참조GetLifecyclePolicy의 섹션을 참조하세요. AWS CLI

### get-object

다음 코드 예시에서는 get-object을 사용하는 방법을 보여 줍니다.

AWS CLI

객체를 다운로드하려면

다음 get-object 예제에서는 지정된 엔드포인트에 객체를 다운로드합니다.

```
aws mediastore-data get-object \
    --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com \
    --path=/folder_name/README.md README.md
```

출력:

```
{
    "ContentLength": "2307346",
    "ContentType": "image/jpeg",
    "LastModified": "Fri, 19 Jul 2019 21:32:20 GMT",
    "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3",
    "StatusCode": 200
}
```

객체의 일부를 다운로드하려면

다음 get-object 예제에서는 객체의 일부를 지정된 엔드포인트에 다운로드합니다.

```
aws mediastore-data get-object \
    --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com \
    --path /folder_name/README.md \
    --range="bytes=0-100" README2.md
```

출력:

```
{
    "StatusCode": 206,
    "ContentRange": "bytes 0-100/2307346",
    "ContentLength": "101",
    "LastModified": "Fri, 19 Jul 2019 21:32:20 GMT",
    "ContentType": "image/jpeg",
    "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3"
```

```
}
```

자세한 내용은 AWS 요소 MediaStore 사용 설명서의 객체 다운로드를 참조하세요.

- 자세한 API 내용은 명령 참조GetObject의 섹션을 참조하세요. AWS CLI

### `list-containers`

다음 코드 예시에서는 `list-containers`을 사용하는 방법을 보여 줍니다.

AWS CLI

컨테이너 목록을 보려면

다음 `list-containers` 예제에서는 계정과 연결된 모든 컨테이너의 목록을 표시합니다.

```
aws mediastore list-containers
```

출력:

```
{
    "Containers": [
        {
            "CreationTime": 1505317931,
            "Endpoint": "https://aaabbbcccdddee.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"
        }
    ]
```

```
}
```

자세한 내용은 AWS 요소 MediaStore 사용 설명서의 컨테이너 목록 보기를 참조하세요.

- 자세한 API 내용은 명령 참조ListContainers의 섹션을 참조하세요. AWS CLI

### `list-items`

다음 코드 예시에서는 `list-items`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 특정 컨테이너에서 객체 및 폴더 목록을 보는 방법

다음 `list-items` 예제에서는 지정된 컨테이너에 저장된 항목(객체 및 폴더)을 표시합니다.

```
aws mediastore-data list-items \
    --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com
```

출력:

```
{
    "Items": [
        {
            "ContentType": "image/jpeg",
            "LastModified": 1563571859.379,
            "Name": "filename.jpg",
            "Type": "OBJECT",
            "ETag":
 "543ab21abcd1a234ab123456a1a2b12345ab12abc12a1234abc1a2bc12345a12",
            "ContentLength": 3784
        },
        {
            "Type": "FOLDER",
            "Name": "ExampleLiveDemo"
        }
    ]
}
```

예제 2: 특정 폴더의 객체 및 폴더 목록을 보려면

다음 `list-items` 예제에서는 특정 폴더에 저장된 항목(객체 및 폴더)을 보여줍니다.

```
aws mediastore-data list-items \
    --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com
```

출력:

```
{
    "Items": [
        {
            "ContentType": "image/jpeg",
            "LastModified": 1563571859.379,
            "Name": "filename.jpg",
            "Type": "OBJECT",
            "ETag":
 "543ab21abcd1a234ab123456a1a2b12345ab12abc12a1234abc1a2bc12345a12",
            "ContentLength": 3784
        },
        {
            "Type": "FOLDER",
            "Name": "ExampleLiveDemo"
        }
    ]
}
```

자세한 내용은 AWS 요소 MediaStore 사용 설명서의 객체 목록 보기를 참조하세요.

- 자세한 API 내용은 명령 참조ListItems의 섹션을 참조하세요. AWS CLI

### list-tags-for-resource

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

컨테이너의 태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 지정된 컨테이너에 할당된 태그 키와 값을 표시합니다.

```
aws mediastore list-tags-for-resource \
    --resource arn:aws:mediastore:us-west-2:1213456789012:container/ExampleContainer
```

출력:

```
{
    "Tags": [
        {
            "Value": "Test",
            "Key": "Environment"
        },
        {
            "Value": "West",
            "Key": "Region"
        }
    ]
}
```

자세한 내용은 요소 참조ListTagsForResource의 섹션을 참조하세요. AWS MediaStore API

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### put-container-policy

다음 코드 예시에서는 put-container-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

컨테이너 정책을 편집하려면

다음 put-container-policy 예제에서는 지정된 컨테이너에 다른 정책을 할당합니다. 이 예제에서는 업데이트된 정책이 라는 파일에 정의되어 있습니다LiveEventsContainerPolicy.json.

```
aws mediastore put-container-policy \
    --container-name LiveEvents \
    --policy file://LiveEventsContainerPolicy.json
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS 요소 MediaStore 사용 설명서의 컨테이너 정책 편집을 참조하세요.

- 자세한 API 내용은 명령 참조PutContainerPolicy의 섹션을 참조하세요. AWS CLI

### put-cors-policy

다음 코드 예시에서는 put-cors-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: CORS 정책 추가

다음 put-cors-policy 예제에서는 교차 오리진 리소스 공유(CORS) 정책을 지정된 컨테이너에 추가합니다. CORS 정책의 내용은 라는 파일에 있습니다corsPolicy.json.

```
aws mediastore put-cors-policy \
    --container-name ExampleContainer \
    --cors-policy file://corsPolicy.json
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 요소 사용 설명서의 컨테이너에 CORS 정책 추가를 참조하세요. AWS MediaStore

예제 2: CORS 정책을 편집하려면

다음 put-cors-policy 예제에서는 지정된 컨테이너에 할당된 크로스 오리진 리소스 공유(CORS) 정책을 업데이트합니다. 업데이트된 CORS 정책의 내용은 라는 파일에 있습니다corsPolicy2.json.

자세한 내용은 AWS 요소 MediaStore 사용 설명서의 CORS 정책 편집을 참조하세요.

- 자세한 API 내용은 명령 참조PutCorsPolicy의 섹션을 참조하세요. AWS CLI

### put-lifecycle-policy

다음 코드 예시에서는 put-lifecycle-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

객체 수명 주기 정책을 생성하려면

다음 put-lifecycle-policy 예제에서는 객체 수명 주기 정책을 지정된 컨테이너에 연결합니다. 이렇게 하면 서비스가 컨테이너에 객체를 저장할 기간을 지정할 수 있습니다. 는 정책에 표시된 대로 만료 날짜에 도달하면 라는 이름의 파일에 있는 컨테이너의 객체를 MediaStore 삭제합니다LiveEventsLifecyclePolicy.json.

```
aws mediastore put-lifecycle-policy \
    --container-name ExampleContainer \
    --lifecycle-policy file://ExampleLifecyclePolicy.json
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 요소 사용 설명서의 컨테이너에 객체 수명 주기 정책 추가를 참조하세요. AWS MediaStore

- 자세한 API 내용은 명령 참조PutLifecyclePolicy의 섹션을 참조하세요. AWS CLI

### put-object

다음 코드 예시에서는 put-object을 사용하는 방법을 보여 줍니다.

AWS CLI

객체를 업로드하려면

다음 put-object 예제에서는 지정된 컨테이너에 객체를 업로드합니다. 컨테이너 내에 객체를 저장할 폴더 경로를 지정할 수 있습니다. 폴더가 이미 있는 경우 AWS Elemental은 객체를 폴더에 MediaStore 저장합니다. 폴더가 없는 경우 서비스는 폴더를 생성한 다음 객체를 폴더에 저장합니다.

```
aws mediastore-data put-object \
    --endpoint https://aaabbbcccdddee.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
```

출력:

```
{
    "ContentSHA256":
 "74b5fdb517f423ed750ef214c44adfe2be36e37d861eafe9c842cbe1bf387a9d",
    "StorageClass": "TEMPORAL",
    "ETag": "af3e4731af032167a106015d1f2fe934e68b32ed1aa297a9e325f5c64979277b"
}
```

자세한 내용은 AWS 요소 MediaStore 사용 설명서의 객체 업로드를 참조하세요.

- 자세한 API 내용은 명령 참조PutObject의 섹션을 참조하세요. AWS CLI

### start-access-logging

다음 코드 예시에서는 start-access-logging을 사용하는 방법을 보여 줍니다.

AWS CLI

컨테이너에서 액세스 로깅을 활성화하려면

다음 `start-access-logging` 예제에서는 지정된 컨테이너에 대한 액세스 로깅을 활성화합니다.

```
aws mediastore start-access-logging \
    --container-name LiveEvents
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 요소 사용 설명서의 컨테이너에 대한 액세스 로깅 활성화를 참조하세요. AWS MediaStore

- 자세한 API 내용은 명령 참조StartAccessLogging의 섹션을 참조하세요. AWS CLI

### `stop-access-logging`

다음 코드 예시에서는 `stop-access-logging`을 사용하는 방법을 보여 줍니다.

AWS CLI

컨테이너에서 액세스 로깅을 비활성화하려면

다음 `stop-access-logging` 예제에서는 지정된 컨테이너에 대한 액세스 로깅을 비활성화합니다.

```
aws mediastore stop-access-logging \
    --container-name LiveEvents
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 요소 사용 설명서의 컨테이너에 대한 액세스 로깅 비활성화를 참조하세요. AWS MediaStore

- 자세한 API 내용은 명령 참조StopAccessLogging의 섹션을 참조하세요. AWS CLI

### `tag-resource`

다음 코드 예시에서는 `tag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

컨테이너에 태그를 추가하려면

다음 `tag-resource` 예제에서는 태그 키와 값을 지정된 컨테이너에 추가합니다.

```
aws mediastore tag-resource \
    --resource arn:aws:mediastore:us-west-2:123456789012:container/ExampleContainer
 \
    --tags '[{"Key": "Region", "Value": "West"}, {"Key": "Environment", "Value":
 "Test"}]'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 요소 참조TagResource의 섹션을 참조하세요. AWS MediaStore API

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### `untag-resource`

다음 코드 예시에서는 `untag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

컨테이너에서 태그를 제거하려면

다음 `untag-resource` 예제에서는 컨테이너에서 지정된 태그 키와 관련 값을 제거합니다.

```
aws mediastore untag-resource \
    --resource arn:aws:mediastore:us-west-2:123456789012:container/ExampleContainer
 \
    --tag-keys Region
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 요소 참조UntagResource의 섹션을 참조하세요. AWS MediaStore API

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

## 를 사용한 Amazon EMR 예제 AWS CLI

다음 코드 예제에서는 Amazon 와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다EMR.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### add-instance-fleet

다음 코드 예시에서는 add-instance-fleet을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터에 태스크 인스턴스 플릿을 추가하려면

이 예제에서는 지정된 클러스터에 새 태스크 인스턴스 플릿을 추가합니다.

명령:

```
aws emr add-instance-fleet --cluster-id 'j-12ABCDEFGHI34JK' --instance-fleet
 InstanceFleetType=TASK,TargetSpotCapacity=1,LaunchSpecifications={SpotSpecification='{Timeo
```

출력:

```
{
   "ClusterId": "j-12ABCDEFGHI34JK",
   "InstanceFleetId": "if-23ABCDEFGHI45JJ"
}
```

- 자세한 API 내용은 명령 참조AddInstanceFleet의 섹션을 참조하세요. AWS CLI

### add-steps

다음 코드 예시에서는 add-steps을 사용하는 방법을 보여 줍니다.

AWS CLI

1: 클러스터에 사용자 지정 JAR 단계를 추가하려면

명령:

```
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
```

필수 파라미터:

```
Jar
```

선택적 파라미터:

```
Type, Name, ActionOnFailure, Args
```

출력:

```
{
    "StepIds":[
        "s-XXXXXXXX",
        "s-YYYYYYYY"
    ]
}
```

2. 클러스터에 스트리밍 단계를 추가하려면

명령:

```
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]
```

필수 파라미터:

```
Type, Args
```

선택적 파라미터:

```
Name, ActionOnFailure
```

JSON 동급(Step.json의 콘텐츠):

```
 [
  {
    "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 인수에는 목록의 고유한 항목으로 옵션과 값이 포함되어야 합니다.

명령(Step.json 사용):

```
aws emr add-steps --cluster-id j-XXXXXXXX --steps file://./step.json
```

출력:

```
{
    "StepIds":[
        "s-XXXXXXXX",
        "s-YYYYYYYY"
    ]
}
```

3. 클러스터에 여러 파일이 있는 스트리밍 단계를 추가하려면(JSON만 해당)

JSON (multiplefiles.json):

```
[
  {
     "Name": "JSON Streaming Step",
     "Type": "STREAMING",
     "ActionOnFailure": "CONTINUE",
     "Args": [
```

```
        "-files",
        "s3://mybucket/mapper.py,s3://mybucket/reducer.py",
        "-mapper",
        "mapper.py",
        "-reducer",
        "reducer.py",
        "-input",
        "s3://mybucket/input",
        "-output",
        "s3://mybucket/output"]
  }
]
```

명령:

```
aws emr add-steps --cluster-id j-XXXXXXXX  --steps file://./multiplefiles.json
```

필수 파라미터:

```
Type, Args
```

선택적 파라미터:

```
Name, ActionOnFailure
```

출력:

```
{
    "StepIds":[
        "s-XXXXXXXX",
    ]
}
```

4. 클러스터에 Hive 단계를 추가하려면

명령:

```
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]
```

필수 파라미터:

```
Type, Args
```

선택적 파라미터:

```
Name, ActionOnFailure
```

출력:

```
{
    "StepIds":[
        "s-XXXXXXXX",
        "s-YYYYYYYY"
    ]
}
```

5. 클러스터에 피그 단계를 추가하려면

명령:

```
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]
```

필수 파라미터:

```
Type, Args
```

선택적 파라미터:

```
Name, ActionOnFailure
```

출력:

```
{
    "StepIds":[
        "s-XXXXXXXX",
        "s-YYYYYYYY"
    ]
}
```

6. 클러스터에 Impala 단계를 추가하려면

명령:

```
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
```

필수 파라미터:

```
Type, Args
```

선택적 파라미터:

```
Name, ActionOnFailure
```

출력:

```
{
    "StepIds":[
        "s-XXXXXXXX",
        "s-YYYYYYYY"
    ]
}
```

- 자세한 API 내용은 명령 참조AddSteps의 섹션을 참조하세요. AWS CLI

### add-tags

다음 코드 예시에서는 add-tags을 사용하는 방법을 보여 줍니다.

AWS CLI

1: 클러스터에 태그를 추가하려면

명령:

```
aws emr add-tags --resource-id j-xxxxxxx --tags name="John Doe" age=29 sex=male
 address="123 East NW Seattle"
```

출력:

```
None
```

2. 클러스터의 태그를 나열하려면

--명령:

```
aws emr describe-cluster --cluster-id j-XXXXXXYY --query Cluster.Tags
```

출력:

```
[
    {
        "Value": "male",
        "Key": "sex"
    },
    {
        "Value": "123 East NW Seattle",
        "Key": "address"
    },
    {
        "Value": "John Doe",
        "Key": "name"
    },
    {
        "Value": "29",
        "Key": "age"
    }
]
```

- 자세한 API 내용은 명령 참조AddTags의 섹션을 참조하세요. AWS CLI

### create-cluster-examples

다음 코드 예시에서는 create-cluster-examples을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 예제의 대부분은 Amazon EMR 서비스 역할 및 Amazon EC2 인스턴스 프로파일을 지정했다고 가정합니다. 이렇게 하지 않은 경우 각 필수 IAM 역할을 지정하거나 클러스터를 생성할 때 `--use-default-roles` 파라미터를 사용해야 합니다. IAM 역할 지정에 대한 자세한 내용은 Amazon 관리 안내서의 AWS 서비스에 대한 Amazon EMR 권한에 대한 IAM 역할 구성을 참조하세요. EMR

예제 1: 클러스터 생성

다음 `create-cluster` 예제에서는 간단한 EMR 클러스터를 생성합니다.

```
aws emr create-cluster \
    --release-label emr-5.14.0 \
    --instance-type m4.large \
    --instance-count 2
```

이 명령은 출력을 생성하지 않습니다.

예제 2: 기본 ServiceRole 및 InstanceProfile 역할로 Amazon EMR 클러스터를 생성하려면

다음 `create-cluster` 예제에서는 `--instance-groups` 구성을 사용하는 Amazon EMR 클러스터를 생성합니다.

```
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
```

예제 3: 인스턴스 플릿을 사용하는 Amazon EMR 클러스터 생성

다음 `create-cluster` 예제에서는 `--instance-fleets` 구성을 사용하는 Amazon EMR 클러스터를 생성하여 각 플릿에 대해 두 개의 인스턴스 유형과 두 개의 EC2 서브넷을 지정합니다.

```
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
```

예제 4: 기본 역할이 있는 클러스터를 생성하려면

다음 `create-cluster` 예제에서는 `--use-default-roles` 파라미터를 사용하여 기본 서비스 역할 및 인스턴스 프로파일을 지정합니다.

```
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
```

예제 5: 클러스터를 생성하고 설치할 애플리케이션을 지정하려면

다음 `create-cluster` 예제에서는 `--applications` 파라미터를 사용하여 Amazon이 EMR 설치하는 애플리케이션을 지정합니다. 이 예제에서는 Hadoop, Hive 및 Pig를 설치합니다.

```
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
```

예제 6: Spark를 포함하는 클러스터를 생성하려면

다음 예제에서는 Spark를 설치합니다.

```
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
```

예제 7: 클러스터 인스턴스에 AMI 사용할 사용자 지정 지정

다음 `create-cluster` 예제에서는 AMI ID가 인 Amazon Linux를 기반으로 클러스터 인스턴스를 생성합니다`ami-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
```

예제 8: 애플리케이션 구성을 사용자 지정하려면

다음 예제에서는 `--configurations` 파라미터를 사용하여 Hadoop에 대한 애플리케이션 사용자 지정이 포함된 JSON 구성 파일을 지정합니다. 자세한 내용은 Amazon EMR 릴리스 가이드의 애플리케이션 구성을 참조하세요.

`configurations.json`의 콘텐츠:

```
[
    {
       "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"
                }
            }
        ]
    }
```

```
]
```

다음 예제에서는 configurations.json 를 로컬 파일로 참조합니다.

```
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
```

다음 예제에서는 Amazon S3의 configurations.json 파일을 참조합니다.

```
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
```

예제 9: 마스터, 코어 및 태스크 인스턴스 그룹을 사용하여 클러스터를 생성하려면

다음 create-cluster 예제에서는 --instance-groups를 사용하여 마스터, 코어 및 태스크 EC2 인스턴스 그룹에 사용할 인스턴스의 유형과 수를 지정합니다.

```
aws emr create-cluster \
    --release-label emr-5.9.0 \
    --instance-
groups Name=Master,InstanceGroupType=MASTER,InstanceType=m4.large,InstanceCount=1 Name=Core,
```

예제 10: 모든 단계를 완료한 후 클러스터를 종료하도록 지정하려면

다음 create-cluster 예제에서는 --auto-terminate를 사용하여 모든 단계를 완료한 후 클러스터를 자동으로 종료하도록 지정합니다.

```
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
```

예제 11: Amazon EC2 키 페어, 네트워크 구성 및 보안 그룹과 같은 클러스터 구성 세부 정보를 지정하려면

다음 `create-cluster` 예제에서는 이름이 인 Amazon EC2 키 페어myKey와 이름이 인 사용자 지정 인스턴스 프로파일을 사용하여 클러스터를 생성합니다myProfile. 키 페어는 클러스터 노드에 대한 SSH 연결을 승인하는 데 사용되며, 대부분 마스터 노드입니다. 자세한 내용은 Amazon 관리 안내서의 SSH 자격 증명에 Amazon EC2 키 페어 사용을 참조하세요. EMR

```
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
```

다음 예제에서는 Amazon VPC 서브넷에 클러스터를 생성합니다.

```
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
```

다음 예제에서는 `us-east-1b` 가용 영역에 클러스터를 생성합니다.

```
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
```

다음 예제에서는 클러스터를 생성하고 Amazon EMR관리형 보안 그룹만 지정합니다.

```
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
```

다음 예제에서는 클러스터를 생성하고 추가 Amazon EC2 보안 그룹만 지정합니다.

```
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
```

다음 예제에서는 클러스터를 생성하고 관리EMR형 보안 그룹과 추가 보안 그룹을 지정합니다.

```
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
```

다음 예제에서는 VPC 프라이빗 서브넷에 클러스터를 생성하고 특정 Amazon EC2 보안 그룹을 사용하여 프라이빗 서브넷의 클러스터에 필요한 Amazon EMR 서비스 액세스를 활성화합니다.

```
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
```

다음 예제에서는 로컬에 ec2_attributes.json 저장된 라는 JSON 파일을 사용하여 보안 그룹 구성 파라미터를 지정합니다. NOTE: JSON 인수에는 목록의 고유한 항목으로 옵션과 값이 포함되어야 합니다.

```
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
```

ec2_attributes.json의 콘텐츠:

```
[
    {
        "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"]
    }
]
```

예제 12: 디버깅을 활성화하고 로그를 지정하려면 URI

다음 create-cluster 예제에서는 --enable-debugging 파라미터를 사용하여 Amazon EMR 콘솔의 디버깅 도구를 사용하여 로그 파일을 더 쉽게 볼 수 있습니다. --log-uri 파라미터는 에 필요합니다--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
```

예제 13: 클러스터를 생성할 때 태그를 추가하려면

태그는 클러스터를 식별하고 관리하는 데 도움이 되는 키-값 페어입니다. 다음 create-cluster 예제에서는 --tags 파라미터를 사용하여 클러스터에 대한 세 개의 태그를 생성합니다. 하나는 키 이름과 name 값이 Shirley Rodriguez이고, 다른 하나는 키 이름과 값이 age이고, 다른 하나는 키 이름과 값이 이고29, 세 번째 태그는 키 department 이름과 값이 입니다Analytics.

```
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
```

다음 예제에서는 클러스터에 적용된 태그를 나열합니다.

```
aws emr describe-cluster \
    --cluster-id j-XXXXXXYY \
    --query Cluster.Tags
```

예제 14: 암호화 및 기타 보안 기능을 활성화하는 보안 구성을 사용하려면

다음 create-cluster 예제에서는 --security-configuration 파라미터를 사용하여 EMR 클러스터에 대한 보안 구성을 지정합니다. Amazon EMR 버전 4.8.0 이상에서 보안 구성을 사용할 수 있습니다.

```
aws emr create-cluster \
    --instance-type m4.large \
    --release-label emr-5.9.0 \
    --security-configuration mySecurityConfiguration
```

예제 15: 인스턴스 그룹에 대해 구성된 추가 EBS 스토리지 볼륨이 있는 클러스터를 생성하려면

추가 EBS 볼륨을 지정할 때는 가 지정된 SizeInGB 경우 VolumeType라는 인수EbsBlockDeviceConfigs가 필요합니다.

다음 create-cluster 예제에서는 코어 인스턴스 그룹의 EC2 인스턴스에 연결된 EBS 볼륨이 여러 개인 클러스터를 생성합니다.

```
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
```

다음 예제에서는 마스터 인스턴스 그룹의 EC2 인스턴스에 연결된 여러 EBS 볼륨이 있는 클러스터를 생성합니다.

```
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}]}' InstanceGrou
 \
    --auto-terminate
```

예제 16: 자동 조정 정책을 사용하여 클러스터를 생성하려면

Amazon EMR 버전 4.0 이상을 사용하여 코어 및 태스크 인스턴스 그룹에 자동 조정 정책을 연결할 수 있습니다. 자동 조정 정책은 Amazon CloudWatch 지표에 대한 응답으로 EC2 인스턴스를 동적으로 추가 및 제거합니다. 자세한 내용은 Amazon  EMR 관리 안내서의 Amazon EMR <https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-automatic-scaling.html>`_에서 자동 크기 조정 사용을 참조하세요.

자동 조정 정책을 연결할 때 를 사용하여 자동 조정을 위한 기본 역할도 지정해야 합니다`--auto-scaling-role EMR_AutoScaling_DefaultRole`.

다음 `create-cluster` 예제에서는 확장 정책 구성을 지정하는 임베디드 JSON 구조의 `AutoScalingPolicy` 인수를 사용하여 `CORE` 인스턴스 그룹에 대한 자동 확장 정책을 지정합니다. 임베디드 JSON 구조가 있는 인스턴스 그룹에는 전체 인수 모음이 작은따옴표로 묶여 있어야 합니다. 포함된 JSON 구조가 없는 인스턴스 그룹의 경우 작은따옴표를 사용하는 것은 선택 사항입니다.

```
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
```

다음 예제에서는 JSON 파일 를 사용하여 클러스터의 모든 인스턴스 그룹의 구성을 instancegroupconfig.json지정합니다. JSON 파일은 코어 인스턴스 그룹에 대한 자동 조정 정책 구성을 지정합니다.

```
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
```

instancegroupconfig.json의 콘텐츠:

```
[
    {
        "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}"
                                }
                            ]
                        }
                    }
                }
            ]
        }
    }
]
```

예제 17: 클러스터를 생성할 때 사용자 지정 JAR 단계 추가

다음 `create-cluster` 예제에서는 Amazon S3에 저장된 JAR 파일을 지정하여 단계를 추가합니다. 단계는 클러스터에 작업을 제출합니다. JAR 파일에 정의된 기본 함수는 EC2 인스턴스가 프로비저닝되고 부트스트랩 작업이 실행되고 애플리케이션이 설치된 후 실행됩니다. 단계는 를 사용하여 지정됩니다`Type=CUSTOM_JAR`.

사용자 지정 JAR 단계에는 의 경로와 파일 이름을 지정하는 `Jar=` 파라미터가 필요합니다JAR. 선택적 파라미터는 `Type`, `Name`, `Args`, `ActionOnFailure`및 입니다`MainClass`. 기본 클래스가 지정되지 않은 경우 JAR 파일은 매`Main-Class`니페스트 파일에 지정해야 합니다.

```
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=s
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
```

예제 18: 클러스터를 생성할 때 스트리밍 단계를 추가하려면

다음 create-cluster 예제에서는 모든 단계가 실행된 후 종료되는 스트리밍 단계를 클러스터에 추가합니다. 스트리밍 단계에는 파라미터 Type 및 가 필요합니다Args. 스트리밍 단계 선택적 파라미터는 Name 및 입니다ActionOnFailure.

다음 예제에서는 단계를 인라인으로 지정합니다.

```
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
```

다음 예제에서는 라는 로컬 저장 JSON 구성 파일을 사용합니다multiplefiles.json. JSON 구성은 여러 파일을 지정합니다. 한 단계에서 여러 파일을 지정하려면 JSON 구성 파일을 사용하여 단계를 지정해야 합니다. JSON 인수에는 옵션과 값이 목록에 자체 항목으로 포함되어야 합니다.

```
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
```

multiplefiles.json의 콘텐츠:

```
[
    {
        "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"
    }
]
```

예제 19: 클러스터 생성 시 Hive 단계 추가

다음 예에서는 클러스터를 생성할 때 Hive 단계를 추가합니다. Hive 단계에는 파라미터 Type 및 가 필요합니다Args. Hive 단계 선택적 파라미터는 Name 및 입니다ActionOnFailure.

```
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
```

예제 20: 클러스터를 생성할 때 피그 단계를 추가하려면

다음 예제에서는 클러스터를 생성할 때 피그 단계를 추가합니다. 피그 단계 필수 파라미터는 Type 및 입니다Args. Pig 단계 선택적 파라미터는 Name 및 입니다ActionOnFailure.

```
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
```

예제 21: 부트스트랩 작업 추가

다음 create-cluster 예제에서는 Amazon S3에 저장된 스크립트로 정의된 두 부트스트랩 작업을 실행합니다.

```
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
```

예제 22: EMRFS 일관된 보기를 활성화하고 RetryCount 및 RetryPeriod 설정을 사용자 지정하려면

다음 create-cluster 예제에서는 EMRFS 일관된 보기를 위해 재시도 횟수와 재시도 기간을 지정합니다. Consistent=true 인수는 필수입니다.

```
aws emr create-cluster \
    --instance-type m4.large \
    --release-label emr-5.9.0 \
    --emrfs Consistent=true,RetryCount=6,RetryPeriod=30
```

다음 예제에서는 라는 로컬 저장 EMRFS 구성 파일을 사용하여 이전 예제와 동일한 JSON 구성을 지정합니다emrfsconfig.json.

```
aws emr create-cluster \
    --instance-type m4.large \
    --release-label emr-5.9.0 \
    --emrfs file://emrfsconfig.json
```

emrfsconfig.json의 콘텐츠:

```
{
    "Consistent": true,
    "RetryCount": 6,
    "RetryPeriod": 30
}
```

예제 23: Kerberos가 구성된 클러스터 생성

다음 `create-cluster` 예제에서는 Kerberos가 활성화된 보안 구성을 사용하여 클러스터를 생성하고 를 사용하여 클러스터에 대한 Kerberos 파라미터를 설정합니다`--kerberos-attributes`.

다음 명령은 클러스터 인라인에 대한 Kerberos 속성을 지정합니다.

```
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
```

다음 명령은 동일한 속성을 지정하지만 라는 로컬 저장 JSON 파일을 참조합니다`kerberos_attributes.json`. 이 예에서 파일은 명령을 실행하는 디렉터리와 동일한 디렉터리에 저장됩니다. Amazon S3에 저장된 구성 파일을 참조할 수도 있습니다.

```
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
```

`kerberos_attributes.json`의 콘텐츠:

```
{
    "Realm": "EC2.INTERNAL",
    "KdcAdminPassword": "123",
    "CrossRealmTrustPrincipalPassword": "123",
}
```

다음 create-cluster 예제에서는 --instance-groups 구성을 사용하고 관리형 크기 조정 정책이 있는 Amazon EMR 클러스터를 생성합니다.

```
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}'
```

다음 create-cluster 예제에서는 '--log-encryption-kms-keyid'를 사용하여 로그 암호화에 사용되는 KMS 키 ID를 정의하는 Amazon EMR 클러스터를 생성합니다.

```
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
```

다음 create-cluster 예제에서는 '--placement-group-configs' 구성을 사용하여 SPREAD 배치 전략을 사용하여 EC2 배치 그룹 내의 고가용성(HA) 클러스터에 마스터 노드를 배치EMR하는 Amazon 클러스터를 생성합니다.

```
aws emr create-cluster \
    --release-label emr-5.30.0 \
    --service-role EMR_DefaultRole \
    --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
    --instance-
groups InstanceGroupType=MASTER,InstanceCount=3,InstanceType=m4.largeInstanceGroupType=CORE,
 \
    --placement-group-configs InstanceRole=MASTER
```

다음 create-cluster 예제에서는 '--auto-termination-policy' 구성을 사용하여 EMR 클러스터에 대한 자동 유휴 종료 임계값을 배치하는 Amazon 클러스터를 생성합니다.

```
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
```

다음 create-cluster 예제에서는 '--os-release-label'를 사용하여 EMR 클러스터 시작을 위한 Amazon Linux 릴리스를 정의하는 Amazon 클러스터를 생성합니다.

```
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
```

예제 24: EBS 루트 볼륨 속성 지정: EMR 릴리스 6.15.0 이상으로 생성된 클러스터 인스턴스의 크기, iops 및 처리량

다음 create-cluster 예제에서는 루트 볼륨 속성을 사용하여 EC2 인스턴스에 대한 루트 볼륨 사양을 구성하는 Amazon EMR 클러스터를 생성합니다.

```
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
```

- 자세한 API 내용은 명령 참조CreateClusterExamples의 섹션을 참조하세요. AWS CLI

### create-default-roles

다음 코드 예시에서는 create-default-roles을 사용하는 방법을 보여 줍니다.

AWS CLI

1: 에 대한 기본 IAM 역할을 생성하려면 EC2

명령:

```
aws emr create-default-roles
```

출력:

```
If the role already exists then the command returns nothing.

If the role does not exist then the output will be:

[
    {
        "RolePolicy": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Action": [
                        "cloudwatch:*",
                        "dynamodb:*",
                        "ec2:Describe*",
                        "elasticmapreduce:Describe*",
                        "elasticmapreduce:ListBootstrapActions",
                        "elasticmapreduce:ListClusters",
                        "elasticmapreduce:ListInstanceGroups",
                        "elasticmapreduce:ListInstances",
                        "elasticmapreduce:ListSteps",
                        "kinesis:CreateStream",
                        "kinesis:DeleteStream",
                        "kinesis:DescribeStream",
                        "kinesis:GetRecords",
                        "kinesis:GetShardIterator",
                        "kinesis:MergeShards",
                        "kinesis:PutRecord",
                        "kinesis:SplitShard",
                        "rds:Describe*",
                        "s3:*",
                        "sdb:*",
                        "sns:*",
                        "sqs:*"
```

```
                ],
                "Resource": "*",
                "Effect": "Allow"
            }
        ]
    },
    "Role": {
        "AssumeRolePolicyDocument": {
            "Version": "2008-10-17",
            "Statement": [
                {
                    "Action": "sts:AssumeRole",
                    "Sid": "",
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "ec2.amazonaws.com"
                    }
                }
            ]
        },
        "RoleId": "AROAIQ5SIQUGL5KMYBJX6",
        "CreateDate": "2015-06-09T17:09:04.602Z",
        "RoleName": "EMR_EC2_DefaultRole",
        "Path": "/",
        "Arn": "arn:aws:iam::176430881729:role/EMR_EC2_DefaultRole"
    }
},
{
    "RolePolicy": {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Action": [
                    "ec2:AuthorizeSecurityGroupIngress",
                    "ec2:CancelSpotInstanceRequests",
                    "ec2:CreateSecurityGroup",
                    "ec2:CreateTags",
                    "ec2:DeleteTags",
                    "ec2:DescribeAvailabilityZones",
                    "ec2:DescribeAccountAttributes",
                    "ec2:DescribeInstances",
                    "ec2:DescribeInstanceStatus",
                    "ec2:DescribeKeyPairs",
                    "ec2:DescribePrefixLists",
```

```
                    "ec2:DescribeRouteTables",
                    "ec2:DescribeSecurityGroups",
                    "ec2:DescribeSpotInstanceRequests",
                    "ec2:DescribeSpotPriceHistory",
                    "ec2:DescribeSubnets",
                    "ec2:DescribeVpcAttribute",
                    "ec2:DescribeVpcEndpoints",
                    "ec2:DescribeVpcEndpointServices",
                    "ec2:DescribeVpcs",
                    "ec2:ModifyImageAttribute",
                    "ec2:ModifyInstanceAttribute",
                    "ec2:RequestSpotInstances",
                    "ec2:RunInstances",
                    "ec2:TerminateInstances",
                    "iam:GetRole",
                    "iam:GetRolePolicy",
                    "iam:ListInstanceProfiles",
                    "iam:ListRolePolicies",
                    "iam:PassRole",
                    "s3:CreateBucket",
                    "s3:Get*",
                    "s3:List*",
                    "sdb:BatchPutAttributes",
                    "sdb:Select",
                    "sqs:CreateQueue",
                    "sqs:Delete*",
                    "sqs:GetQueue*",
                    "sqs:ReceiveMessage"
                ],
                "Resource": "*",
                "Effect": "Allow"
            }
        ]
    },
    "Role": {
        "AssumeRolePolicyDocument": {
            "Version": "2008-10-17",
            "Statement": [
                {
                    "Action": "sts:AssumeRole",
                    "Sid": "",
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "elasticmapreduce.amazonaws.com"
```

```
                        }
                    }
                ]
            },
            "RoleId": "AROAI3SRVPPVSRDLARBPY",
            "CreateDate": "2015-06-09T17:09:10.401Z",
            "RoleName": "EMR_DefaultRole",
            "Path": "/",
            "Arn": "arn:aws:iam::176430881729:role/EMR_DefaultRole"
        }
    }
]
```

- 자세한 API 내용은 명령 참조CreateDefaultRoles의 섹션을 참조하세요. AWS CLI

### `create-security-configuration`

다음 코드 예시에서는 `create-security-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

1: 인증서 공급자의 경우 에서 전송 중 암호화를 활성화하고, S3 암호화의 PEM 경우 -S3로, 로컬 디스크 키 공급자의 경우 AWS-KMS로 저장 중 암호화를 활성화하여 보안 구성을 생성하려면 SSE-S3

명령:

```
 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"
                        }
                }
        }
}'
```

출력:

```
{
"CreationDateTime": 1474070889.129,
"Name": "MySecurityConfig"
}
```

JSON 동급( security_configuration.json의 콘텐츠):

```
{
    "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"
            }
        }
    }
}
```

명령( security_configuration.json 사용):

```
aws emr create-security-configuration --name "MySecurityConfig" --security-
configuration file://./security_configuration.json
```

출력:

```
{
"CreationDateTime": 1474070889.129,
"Name": "MySecurityConfig"
}
```

2. 클러스터 전용 KDC 및 교차 영역 신뢰를 사용하여 Kerberos가 활성화된 보안 구성을 생성하려면

명령:

```
 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"
                 }
             }
         }
     }
}'
```

출력:

```
{
"CreationDateTime": 1490225558.982,
"Name": "MySecurityConfig"
}
```

JSON 동급( security_configuration.json의 콘텐츠):

```
{
    "AuthenticationConfiguration": {
        "KerberosConfiguration": {
            "Provider": "ClusterDedicatedKdc",
            "ClusterDedicatedKdcConfiguration": {
                "TicketLifetimeInHours": 24,
                "CrossRealmTrustConfiguration": {
                    "Realm": "AD.DOMAIN.COM",
                    "Domain": "ad.domain.com",
                    "AdminServer": "ad.domain.com",
                    "KdcServer": "ad.domain.com"
                }
            }
        }
    }
}
```

명령( security_configuration.json 사용):

```
aws emr create-security-configuration --name "MySecurityConfig" --security-
configuration file://./security_configuration.json
```

출력:

```
{
"CreationDateTime": 1490225558.982,
"Name": "MySecurityConfig"
}
```

- 자세한 API 내용은 명령 참조CreateSecurityConfiguration의 섹션을 참조하세요. AWS CLI

### `delete-security-configuration`

다음 코드 예시에서는 `delete-security-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 리전에서 보안 구성을 삭제하려면

명령:

```
aws emr delete-security-configuration --name MySecurityConfig
```

출력:

```
None
```

- 자세한 API 내용은 명령 참조DeleteSecurityConfiguration의 섹션을 참조하세요. AWS CLI

### `describe-cluster`

다음 코드 예시에서는 `describe-cluster`을 사용하는 방법을 보여 줍니다.

AWS CLI

명령:

```
aws emr describe-cluster --cluster-id j-XXXXXXXX
```

출력:

```
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-xxxxxxxx",
                    "EmrManagedMasterSecurityGroup": "sg-xxxxxxxx",
                    "IamInstanceProfile": "EMR_EC2_DefaultRole",
                    "Ec2KeyName": "myKey",
                    "Ec2AvailabilityZone": "us-east-1c",
                    "EmrManagedSlaveSecurityGroup": "sg-yyyyyyyyy"
```

```
    },
    "Name": "My Cluster",
    "ServiceRole": "EMR_DefaultRole",
    "Tags": [],
    "TerminationProtected": true,
    "UnhealthyNodeReplacement": true,
    "ReleaseLabel": "emr-4.0.0",
    "NormalizedInstanceHours": 96,
    "InstanceGroups": [
        {
            "RequestedInstanceCount": 2,
            "Status": {
                "Timeline": {
                    "ReadyDateTime": 1436475074.245,
                    "CreationDateTime": 1436474656.564,
                    "EndDateTime": 1436638158.387
                },
                "State": "RUNNING",
                "StateChangeReason": {
                    "Message": "",
                }
            },
            "Name": "CORE",
            "InstanceGroupType": "CORE",
            "Id": "ig-YYYYYYY",
            "Configurations": [],
            "InstanceType": "m3.large",
            "Market": "ON_DEMAND",
            "RunningInstanceCount": 2
        },
        {
            "RequestedInstanceCount": 1,
            "Status": {
                "Timeline": {
                    "ReadyDateTime": 1436475074.245,
                    "CreationDateTime": 1436474656.564,
                    "EndDateTime": 1436638158.387
                },
                "State": "RUNNING",
                "StateChangeReason": {
                    "Message": "",
                }
            },
            "Name": "MASTER",
```

```
                        "InstanceGroupType": "MASTER",
                        "Id": "ig-XXXXXXXXX",
                        "Configurations": [],
                        "InstanceType": "m3.large",
                        "Market": "ON_DEMAND",
                        "RunningInstanceCount": 1
                    }
                ],
                "Applications": [
                    {
                        "Name": "Hadoop"
                    }
                ],
                "VisibleToAllUsers": true,
                "BootstrapActions": [],
                "MasterPublicDnsName": "ec2-54-147-144-78.compute-1.amazonaws.com",
                "AutoTerminate": false,
                "Id": "j-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"
        }
    }
```

- 자세한 API 내용은 명령 참조DescribeCluster의 섹션을 참조하세요. AWS CLI

### describe-step

다음 코드 예시에서는 describe-step을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 클러스터 ID가 j-3SD91U2E1L2QX인 클러스터에서 단계 ID가 s-3LZC0QUT43AM인 단계를 설명합니다.

```
aws emr describe-step --cluster-id j-3SD91U2E1L2QX --step-id s-3LZC0QUT43AM
```

출력:

```
{
    "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"
    }
}
```

- 자세한 API 내용은 명령 참조DescribeStep의 섹션을 참조하세요. AWS CLI

### get

다음 코드 예시에서는 get을 사용하는 방법을 보여 줍니다.

AWS CLI

다음은 클러스터 ID가 인 클러스터의 마스터 인스턴스에서 hadoop-examples.jar 아카이브를 다운로드합니다j-3SD91U2E1L2QX.

```
aws emr get --cluster-id j-3SD91U2E1L2QX --key-pair-file ~/.ssh/mykey.pem --src /
home/hadoop-examples.jar --dest ~
```

- 자세한 API 내용은 AWS CLI 명령 참조에서 가져오기를 참조하세요.

### list-clusters

다음 코드 예시에서는 list-clusters을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 현재 리전의 모든 활성 EMR 클러스터를 나열합니다.

```
aws emr list-clusters --active
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListClusters의 섹션을 참조하세요. AWS CLI

### `list-instance-fleets`

다음 코드 예시에서는 `list-instance-fleets`을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터에서 인스턴스 플릿의 구성 세부 정보를 가져오려면

이 예제에서는 지정된 클러스터의 인스턴스 플릿에 대한 세부 정보를 나열합니다.

명령:

```
list-instance-fleets --cluster-id 'j-12ABCDEFGHI34JK'
```

출력:

```
{
  "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListInstanceFleets의 섹션을 참조하세요. AWS CLI

### `list-instances`

다음 코드 예시에서는 `list-instances`을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 클러스터 ID가 인 클러스터의 모든 인스턴스를 나열합니다`j-3C6XNQ39VR9WL`.

```
aws emr list-instances --cluster-id j-3C6XNQ39VR9WL
```

출력:

```
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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListInstances의 섹션을 참조하세요. AWS CLI

### `list-security-configurations`

다음 코드 예시에서는 `list-security-configurations`을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 리전의 보안 구성을 나열하려면

명령:

```
aws emr list-security-configurations
```

출력:

```
{
    "SecurityConfigurations": [
        {
            "CreationDateTime": 1473889697.417,
            "Name": "MySecurityConfig-1"
        },
        {
            "CreationDateTime": 1473889697.417,
            "Name": "MySecurityConfig-2"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListSecurityConfigurations의 섹션을 참조하세요. AWS CLI

### `list-steps`

다음 코드 예시에서는 `list-steps`을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 클러스터 ID가 `j-3SD91U2E1L2QX`인 클러스터의 모든 단계를 나열합니다.

```
aws emr list-steps --cluster-id j-3SD91U2E1L2QX
```

- 자세한 API 내용은 명령 참조ListSteps의 섹션을 참조하세요. AWS CLI

### `modify-cluster-attributes`

다음 코드 예시에서는 `modify-cluster-attributes`을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 ID가 인 EMR 클러스터의 가시성을 `j-301CDNY0J5XM4` 모든 사용자에게 설정합니다.

```
aws emr modify-cluster-attributes --cluster-id j-301CDNY0J5XM4 --visible-to-all-
users
```

- 자세한 API 내용은 명령 참조ModifyClusterAttributes의 섹션을 참조하세요. AWS CLI

### modify-instance-fleet

다음 코드 예시에서는 modify-instance-fleet을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스 플릿의 대상 캐패시터를 변경하려면

이 예제에서는 지정된 인스턴스 플릿에 대해 온디맨드 및 스팟 대상 용량을 1로 변경합니다.

명령:

```
aws emr modify-instance-fleet --cluster-id 'j-12ABCDEFGHI34JK' --instance-fleet
 InstanceFleetId='if-2ABC4DEFGHIJ4',TargetOnDemandCapacity=1,TargetSpotCapacity=1
```

- 자세한 API 내용은 명령 참조ModifyInstanceFleet의 섹션을 참조하세요. AWS CLI

### put

다음 코드 예시에서는 put을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 클러스터 IDhealthcheck.sh가 인 클러스터의 마스터 인스턴스에 라는 파일을 업로드합니다j-3SD91U2E1L2QX.

```
aws emr put --cluster-id j-3SD91U2E1L2QX --key-pair-file ~/.ssh/mykey.pem --src ~/
scripts/healthcheck.sh --dest /home/hadoop/bin/healthcheck.sh
```

- 자세한 API 내용은 명령 내 입력 참조를 참조하세요. AWS CLI

### remove-tags

다음 코드 예시에서는 remove-tags을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 클러스터 ID가 인 클러스터prod에서 키가 있는 태그를 제거합니다j-3SD91U2E1L2QX.

```
aws emr remove-tags --resource-id j-3SD91U2E1L2QX --tag-keys prod
```

- 자세한 API 내용은 명령 참조RemoveTags의 섹션을 참조하세요. AWS CLI

### `schedule-hbase-backup`

다음 코드 예시에서는 `schedule-hbase-backup`을 사용하는 방법을 보여 줍니다.

AWS CLI

참고: 이 명령은 AMI 버전 2.x 및 3.xHBase에서만 사용할 수 있습니다.

1: 전체 HBase 백업을 예약하려면 >>>>>>> 06ab6d6e13564b5733d75abaf3b599f93cf39a23

명령:

```
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
```

출력:

```
None
```

2. 증분 HBase 백업을 예약하려면

명령:

```
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
```

출력:

```
None
```

- 자세한 API 내용은 명령 참조ScheduleHbaseBackup의 섹션을 참조하세요. AWS CLI

### `socks`

다음 코드 예시에서는 `socks`을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 클러스터 ID가 인 클러스터의 마스터 인스턴스와 양말 연결을 엽니다j-3SD91U2E1L2QX.

```
aws emr socks --cluster-id j-3SD91U2E1L2QX --key-pair-file ~/.ssh/mykey.pem
```

키 페어 파일 옵션은 프라이빗 키 파일의 로컬 경로를 가져옵니다.

- API 자세한 내용은 AWS CLI 명령 참조의 삭스를 참조하세요.

### ssh

다음 코드 예시에서는 ssh을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 클러스터 ID가 인 클러스터의 마스터 인스턴스와 ssh 연결을 엽니다j-3SD91U2E1L2QX.

```
aws emr ssh --cluster-id j-3SD91U2E1L2QX --key-pair-file ~/.ssh/mykey.pem
```

키 페어 파일 옵션은 프라이빗 키 파일의 로컬 경로를 가져옵니다.

출력:

```
ssh -o StrictHostKeyChecking=no -o ServerAliveInterval=10 -i /home/local/user/.ssh/
mykey.pem hadoop@ec2-52-52-41-150.us-west-2.compute.amazonaws.com
Warning: Permanently added 'ec2-52-52-41-150.us-
west-2.compute.amazonaws.com,52.52.41.150' (ECDSA) to the list of known hosts.
Last login: Mon Jun  1 23:15:38 2015

      __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2015.03-release-notes/
26 package(s) needed for security, out of 39 available
Run "sudo yum update" to apply all updates.

--------------------------------------------------------------------------------
```

```
Welcome to Amazon Elastic MapReduce running Hadoop and Amazon Linux.

Hadoop is installed in /home/hadoop. Log files are in /mnt/var/log/hadoop. Check
/mnt/var/log/hadoop/steps for diagnosing step failures.

The Hadoop UI can be accessed via the following commands:

  ResourceManager    lynx http://ip-172-21-11-216:9026/
  NameNode           lynx http://ip-172-21-11-216:9101/

-------------------------------------------------------------------------------

[hadoop@ip-172-31-16-216 ~]$
```

- API 자세한 내용은 AWS CLI 명령 참조의 Ssh를 참조하세요.

## 를 사용한 Amazon EMR on EKS 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface Amazon EMR on 에서 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다EKS.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### **update-role-trust-policy**

다음 코드 예시에서는 update-role-trust-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 Amazon과 함께 사용할 IAM 역할의 신뢰 정책을 업데이트하려면 EMR EKS

이 예제 명령은 example_cluster라는 클러스터의 example_namespace 네임스페이스와 함께 Amazon on과 함께 사용할 수 있도록 example_iam_role이라는 역할의 신뢰 정책을 업데이트합니다. EMR EKS  EKS

명령:

```
aws emr-containers update-role-trust-policy \
    --cluster example_cluster \
    --namespace example_namespace \
    --role-name example_iam_role
```

출력:

```
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.
```

- 자세한 API 내용은 명령 참조UpdateRoleTrustPolicy의 섹션을 참조하세요. AWS CLI

## EventBridge 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 EventBridge.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### `delete-rule`

다음 코드 예시에서는 `delete-rule`을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudWatch 이벤트 규칙을 삭제하려면

이 예제에서는 라는 규칙을 삭제합니다EC2InstanceStateChanges.

```
aws events delete-rule --name "EC2InstanceStateChanges"
```

- 자세한 API 내용은 명령 참조DeleteRule의 섹션을 참조하세요. AWS CLI

### `describe-rule`

다음 코드 예시에서는 `describe-rule`을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudWatch 이벤트 규칙에 대한 정보를 표시하려면

이 예제에서는 라는 규칙에 대한 정보를 표시합니다 DailyLambdaFunction.

```
aws events describe-rule --name "DailyLambdaFunction"
```

- 자세한 API 내용은 명령 참조DescribeRule의 섹션을 참조하세요. AWS CLI

### `disable-rule`

다음 코드 예시에서는 `disable-rule`을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudWatch 이벤트 규칙을 비활성화하려면

이 예제에서는 라는 규칙을 비활성화합니다 DailyLambdaFunction. 규칙이 삭제되지는 않습니다.

```
aws events disable-rule --name "DailyLambdaFunction"
```

- 자세한 API 내용은 명령 참조DisableRule의 섹션을 참조하세요. AWS CLI

### enable-rule

다음 코드 예시에서는 enable-rule을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudWatch 이벤트 규칙을 활성화하려면

이 예제에서는 이전에 비활성화 DailyLambdaFunction된 이라는 규칙을 활성화합니다.

```
aws events enable-rule --name "DailyLambdaFunction"
```

- 자세한 API 내용은 명령 참조EnableRule의 섹션을 참조하세요. AWS CLI

### list-rule-names-by-target

다음 코드 예시에서는 list-rule-names-by-target을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 대상이 있는 모든 규칙을 표시하는 방법

이 예제에서는 'MyFunctionName'라는 Lambda 함수를 대상으로 하는 모든 규칙을 표시합니다.

```
aws events list-rule-names-by-target --target-arn "arn:aws:lambda:us-
east-1:123456789012:function:MyFunctionName"
```

- 자세한 API 내용은 명령 참조ListRuleNamesByTarget의 섹션을 참조하세요. AWS CLI

### list-rules

다음 코드 예시에서는 list-rules을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 CloudWatch 이벤트 규칙 목록을 표시하려면

이 예제는 리전의 모든 CloudWatch 이벤트 규칙을 표시합니다.

```
aws events list-rules
```

특정 문자열로 시작하는 CloudWatch 이벤트 규칙 목록을 표시합니다.

이 예제에서는 이름이 “일일”로 시작하는 리전의 모든 CloudWatch 이벤트 규칙을 표시합니다.

```
aws events list-rules --name-prefix "Daily"
```

- 자세한 API 내용은 명령 참조ListRules의 섹션을 참조하세요. AWS CLI

### `list-targets-by-rule`

다음 코드 예시에서는 `list-targets-by-rule`을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudWatch 이벤트 규칙의 모든 대상을 표시하려면

이 예제에서는 라는 규칙의 모든 대상을 표시합니다 DailyLambdaFunction.

```
aws events list-targets-by-rule --rule  "DailyLambdaFunction"
```

- 자세한 API 내용은 명령 참조ListTargetsByRule의 섹션을 참조하세요. AWS CLI

### `put-events`

다음 코드 예시에서는 `put-events`을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudWatch 이벤트로 사용자 지정 이벤트를 보내려면

이 예제에서는 사용자 지정 이벤트를 CloudWatch 이벤트로 보냅니다. 이벤트는 putevents.json 파일 내에 포함되어 있습니다.

```
aws events put-events --entries file://putevents.json
```

putevents.json file 파일의 콘텐츠는 다음과 같습니다.

```
[
  {
    "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"
   }
]
```

- 자세한 API 내용은 명령 참조PutEvents의 섹션을 참조하세요. AWS CLI

### put-rule

다음 코드 예시에서는 put-rule을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudWatch 이벤트 규칙을 생성하려면

이 예제에서는 매일 오전 9시()에 트리거되는 규칙을 생성합니다UTC. put-targets를 사용하여 Lambda 함수를 이 규칙의 대상으로 추가하는 경우 매일 지정된 시간에 Lambda 함수를 실행할 수 있습니다.

```
aws events put-rule --name "DailyLambdaFunction" --schedule-expression "cron(0 9 *
 * ? *)"
```

이 예제에서는 리전의 EC2 인스턴스가 상태를 변경할 때 트리거되는 규칙을 생성합니다.

```
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"
```

이 예제에서는 리전의 EC2 인스턴스가 중지되거나 종료될 때 트리거되는 규칙을 생성합니다.

```
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"
```

- 자세한 API 내용은 명령 참조PutRule의 섹션을 참조하세요. AWS CLI

### put-targets

다음 코드 예시에서는 put-targets을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudWatch 이벤트 규칙의 대상을 추가하려면

다음 예시에서는 Lambda 함수를 규칙 대상으로 추가합니다.

```
aws events put-targets --rule DailyLambdaFunction --targets
 "Id"="1","Arn"="arn:aws:lambda:us-east-1:123456789012:function:MyFunctionName"
```

이 예시에서는 Amazon Kinesis 스트림을 대상으로 설정하여 이 규칙에 의해 포착된 이벤트가 스트림으로 전달되도록 합니다.

```
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"
```

이 예시에서는 두 개의 Amazon Kinesis 스트림을 하나의 규칙 대상으로 설정합니다.

```
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"
```

- 자세한 API 내용은 명령 참조PutTargets의 섹션을 참조하세요. AWS CLI

### `remove-targets`

다음 코드 예시에서는 `remove-targets`을 사용하는 방법을 보여 줍니다.

AWS CLI

이벤트 대상을 제거하는 방법

이 예제에서는 MyStream1이라는 Amazon Kinesis 스트림을 규칙의 대상으로서 제거합니다 DailyLambdaFunction. DailyLambdaFunction 이 생성되면 이 스트림은 ID가 Target1인 대상으로 설정되었습니다.

```
aws events remove-targets --rule "DailyLambdaFunction" --ids "Target1"
```

- 자세한 API 내용은 명령 참조RemoveTargets의 섹션을 참조하세요. AWS CLI

### `test-event-pattern`

다음 코드 예시에서는 `test-event-pattern`을 사용하는 방법을 보여 줍니다.

AWS CLI

이벤트 패턴이 지정된 이벤트와 일치하는지 확인하려면

이 예제에서는 패턴 “source:com.mycompany.myapp”이 지정된 이벤트와 일치하는지 테스트합니다. 이 예제에서 출력은 “true”입니다.

```
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\"}"
```

- 자세한 API 내용은 명령 참조TestEventPattern의 섹션을 참조하세요. AWS CLI

## 를 사용한 Firewall Manager 예제 AWS CLI

다음 코드 예제에서는 Firewall Manager AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### `associate-admin-account`

다음 코드 예시에서는 `associate-admin-account`을 사용하는 방법을 보여 줍니다.

AWS CLI

Firewall Manager 관리자 계정을 설정하려면

다음 `associate-admin-account` 예제에서는 Firewall Manager의 관리자 계정을 설정합니다.

```
aws fms associate-admin-account \
    --admin-account 123456789012
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 ,Firewall Manager 및 Shield Advanced Developer Guide의 AWS Firewall Manager 관리자 계정 설정을 참조하세요. AWS WAF AWS AWS

- 자세한 API 내용은 명령 참조AssociateAdminAccount의 섹션을 참조하세요. AWS CLI

### `delete-notification-channel`

다음 코드 예시에서는 `delete-notification-channel`을 사용하는 방법을 보여 줍니다.

AWS CLI

Firewall Manager 로그에 대한 SNS 주제 정보를 제거하려면

다음 `delete-notification-channel` 예제에서는 SNS 주제 정보를 제거합니다.

```
aws fms delete-notification-channel
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 , 방화벽 관리자 및 Shield 고급 개발자 안내서의 Amazon SNS 알림 및 Amazon CloudWatch 경보 구성을 참조하세요. AWS WAF AWS AWS

- 자세한 API 내용은 명령 참조DeleteNotificationChannel의 섹션을 참조하세요. AWS CLI

### `delete-policy`

다음 코드 예시에서는 `delete-policy`을 사용하는 방법을 보여 줍니다.

AWS CLI

Firewall Manager 정책을 삭제하려면

다음 `delete-policy` 예제에서는 지정된 ID가 있는 정책을 모든 리소스와 함께 제거합니다.

```
aws fms delete-policy \
    --policy-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --delete-all-policy-resources
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 ,Firewall Manager 및 Shield Advanced Developer Guide의 AWS Firewall Manager 정책 작업을 참조하세요. AWS WAF AWS AWS

- 자세한 API 내용은 명령 참조DeletePolicy의 섹션을 참조하세요. AWS CLI

### `disassociate-admin-account`

다음 코드 예시에서는 `disassociate-admin-account`을 사용하는 방법을 보여 줍니다.

AWS CLI

Firewall Manager 관리자 계정을 제거하려면

다음 `disassociate-admin-account` 예제에서는 Firewall Manager에서 현재 관리자 계정 연결을 제거합니다.

```
aws fms disassociate-admin-account
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 ,Firewall Manager 및 Shield 고급 개발자 안내서의 AWS Firewall Manager 관리자 계정 설정을 참조하세요. AWS WAF AWS AWS

- 자세한 API 내용은 명령 참조DisassociateAdminAccount의 섹션을 참조하세요. AWS CLI

### get-admin-account

다음 코드 예시에서는 get-admin-account을 사용하는 방법을 보여 줍니다.

AWS CLI

Firewall Manager 관리자 계정을 검색하려면

다음 get-admin-account 예제에서는 관리자 계정을 검색합니다.

```
aws fms get-admin-account
```

출력:

```
{
    "AdminAccount": "123456789012",
    "RoleStatus": "READY"
}
```

자세한 내용은 ,AWS WAF AWS Firewall AWS AWS Manager 및 Shield Advanced Developer Guide의 Firewall Manager 사전 요구 사항을 참조하세요.

- 자세한 API 내용은 명령 참조GetAdminAccount의 섹션을 참조하세요. AWS CLI

### get-compliance-detail

다음 코드 예시에서는 get-compliance-detail을 사용하는 방법을 보여 줍니다.

AWS CLI

계정의 규정 준수 정보를 검색하려면

다음 get-compliance-detail 예제에서는 지정된 정책 및 멤버 계정에 대한 규정 준수 정보를 검색합니다.

```
aws fms get-compliance-detail \
    --policy-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --member-account 123456789012
```

출력:

```
{
    "PolicyComplianceDetail": {
    "EvaluationLimitExceeded": false,
    "IssueInfoMap": {},
    "MemberAccount": "123456789012",
    "PolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "PolicyOwner": "123456789012",
    "Violators": []
}
```

자세한 내용은 ,AWS WAF AWS Firewall Manager 및 AWS Shield Advanced Developer Guide의 정책을 사용한 리소스 규정 준수 보기를 참조하세요.

- 자세한 API 내용은 명령 참조GetComplianceDetail의 섹션을 참조하세요. AWS CLI

### get-notification-channel

다음 코드 예시에서는 get-notification-channel을 사용하는 방법을 보여 줍니다.

AWS CLI

Firewall Manager 로그에 대한 SNS 주제 정보를 검색하려면

다음 get-notification-channel 예제에서는 SNS 주제 정보를 검색합니다.

```
aws fms get-notification-channel
```

출력:

```
{
    "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"
}
```

자세한 내용은 , 방화벽 관리자 및 Shield 고급 개발자 안내서의 Amazon SNS 알림 및 Amazon CloudWatch 경보 구성을 참조하세요. AWS WAF AWS AWS

- 자세한 API 내용은 명령 참조GetNotificationChannel의 섹션을 참조하세요. AWS CLI

### get-policy

다음 코드 예시에서는 get-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

Firewall Manager 정책을 검색하려면

다음 get-policy 예제에서는 지정된 ID로 정책을 검색합니다.

```
aws fms get-policy \
    --policy-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

출력:

```
{
    "Policy": {
        "PolicyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "PolicyName": "test",
        "PolicyUpdateToken": "1:p+2RpKR4wPFx7mcrL1UOQQ==",
        "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"
```

```
}
```

자세한 내용은 ,Firewall Manager 및 Shield 고급 개발자 안내서의 AWS Firewall Manager 정책 작업을 참조하세요. AWS WAF AWS AWS

- 자세한 API 내용은 명령 참조GetPolicy의 섹션을 참조하세요. AWS CLI

### `list-compliance-status`

다음 코드 예시에서는 `list-compliance-status`을 사용하는 방법을 보여 줍니다.

AWS CLI

멤버 계정에 대한 정책 규정 준수 정보를 검색하려면

다음 `list-compliance-status` 예제에서는 지정된 정책에 대한 멤버 계정 규정 준수 정보를 검색합니다.

```
aws fms list-compliance-status \
    --policy-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

출력:

```
{
    "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": {}
        }
    ]
}
```

자세한 내용은 ,AWS WAF AWS Firewall Manager 및 AWS Shield Advanced Developer Guide의 정책을 통한 리소스 규정 준수 보기를 참조하세요.

- 자세한 API 내용은 명령 참조ListComplianceStatus의 섹션을 참조하세요. AWS CLI

### `list-member-accounts`

다음 코드 예시에서는 `list-member-accounts`을 사용하는 방법을 보여 줍니다.

AWS CLI

조직의 멤버 계정을 검색하려면

다음 `list-member-accounts` 예제에서는 Firewall Manager 관리자 조직에 있는 모든 멤버 계정을 나열합니다.

```
aws fms list-member-accounts
```

출력:

```
{
    "MemberAccounts": [
        "222222222222",
        "333333333333",
        "444444444444"
    ]
}
```

자세한 내용은 , 방화벽 AWS 관리자 및 Shield 고급 개발자 안내서의 Firewall Manager를 참조하세요. AWS WAF AWS AWS

- 자세한 API 내용은 명령 참조ListMemberAccounts의 섹션을 참조하세요. AWS CLI

### `list-policies`

다음 코드 예시에서는 `list-policies`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 Firewall Manager 정책을 검색하려면

다음 `list-policies` 예제에서는 계정의 정책 목록을 검색합니다. 이 예제에서 출력은 요청당 2개의 결과로 제한됩니다. 각 호출은 목록에 대한 다음 결과 세트를 가져오기 위해 다음 `list-policies` 호출에서 `--starting-token` 파라미터 값으로 사용할 수 `NextToken` 있는 를 반환합니다.

```
aws fms list-policies \
    --max-items 2
```

출력:

```
{
    "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": "eyJOZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ=="
}
```

자세한 내용은 ,Firewall Manager 및 Shield 고급 개발자 안내서의 AWS Firewall Manager 정책 작업을 참조하세요. AWS WAF AWS AWS

- 자세한 API 내용은 명령 참조ListPolicies의 섹션을 참조하세요. AWS CLI

### put-notification-channel

다음 코드 예시에서는 put-notification-channel을 사용하는 방법을 보여 줍니다.

AWS CLI

Firewall Manager 로그에 대한 SNS 주제 정보를 설정하려면

다음 put-notification-channel 예제에서는 SNS 주제 정보를 설정합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 , 방화벽 관리자 및 Shield 고급 개발자 안내서의 Amazon SNS 알림 및 Amazon CloudWatch 경보 구성을 참조하세요. AWS WAF AWS AWS

- 자세한 API 내용은 명령 참조PutNotificationChannel의 섹션을 참조하세요. AWS CLI

### put-policy

다음 코드 예시에서는 put-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

Firewall Manager 정책을 생성하려면

다음 put-policy 예제에서는 Firewall Manager 보안 그룹 정책을 생성합니다.

```
aws fms put-policy \
    --cli-input-json file://policy.json
```

policy.json의 콘텐츠:

```
{
    "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"
        }
    ]
}
```

출력:

```
{
    "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"
}
```

자세한 내용은 ,Firewall Manager 및 Shield Advanced Developer Guide의 AWS Firewall Manager 정책 작업을 참조하세요. AWS WAF AWS AWS

- 자세한 API 내용은 명령 참조PutPolicy의 섹션을 참조하세요. AWS CLI

# AWS FIS 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS FIS.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `create-experiment-template`

다음 코드 예시에서는 `create-experiment-template`을 사용하는 방법을 보여 줍니다.

AWS CLI

실험 템플릿을 생성하려면

다음 `create-experiment-template` 예제에서는 계정에 AWS FIS 실험 템플릿을 생성합니다.

```
aws fis create-experiment-template \
    --cli-input-json file://myfile.json
```

`myfile.json`의 콘텐츠:

```
{
    "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"
}
```

출력:

```
{
    "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": {}
    }
}
```

자세한 내용은 AWS Fault Injection Simulator 사용 설명서의 실험 템플릿 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateExperimentTemplate의 섹션을 참조하세요. AWS CLI

### delete-experiment-template

다음 코드 예시에서는 delete-experiment-template을 사용하는 방법을 보여 줍니다.

AWS CLI

실험 템플릿을 삭제하려면

다음 delete-experiment-template 예제에서는 지정된 실험 템플릿을 삭제합니다.

```
aws fis delete-experiment-template \
    --id ABCDE1fgHIJkLmNop
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS Fault Injection Simulator 사용 설명서의 실험 템플릿 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteExperimentTemplate의 섹션을 참조하세요. AWS CLI

### `get-action`

다음 코드 예시에서는 `get-action`을 사용하는 방법을 보여 줍니다.

AWS CLI

작업 세부 정보를 가져오려면

다음 `get-action` 예제에서는 지정된 작업의 세부 정보를 가져옵니다.

```
aws fis get-action \
```

```
    --id aws:ec2:stop-instances
```

출력:

```
{
    "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": {}
    }
}
```

자세한 내용은 AWS Fault Injection Simulator 사용 설명서의 작업을 참조하세요.

- 자세한 API 내용은 명령 참조GetAction의 섹션을 참조하세요. AWS CLI

### get-experiment-template

다음 코드 예시에서는 get-experiment-template을 사용하는 방법을 보여 줍니다.

AWS CLI

실험 템플릿 세부 정보를 가져오려면

다음 get-experiment-template 예제에서는 지정된 실험 템플릿의 세부 정보를 가져옵니다.

```
aws fis get-experiment-template \
    --id ABCDE1fgHIJkLmNop
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 AWS Fault Injection Simulator 사용 설명서의 실험 템플릿을 참조하세요.

- 자세한 API 내용은 명령 참조GetExperimentTemplate의 섹션을 참조하세요. AWS CLI

### get-experiment

다음 코드 예시에서는 get-experiment을 사용하는 방법을 보여 줍니다.

AWS CLI

실험 세부 정보를 가져오려면

다음 get-experiment 예제에서는 지정된 실험의 세부 정보를 가져옵니다.

```
aws fis get-experiment \
    --id ABC12DeFGhI3jKLMNOP
```

출력:

```
{
    "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": {}
    }
}
```

자세한 내용은 AWS Fault Injection Simulator 사용 설명서의 실험 AWS FIS을 참조하세요.

- 자세한 API 내용은 명령 참조GetExperiment의 섹션을 참조하세요. AWS CLI

### `list-actions`

다음 코드 예시에서는 `list-actions`을 사용하는 방법을 보여 줍니다.

AWS CLI

작업을 나열하려면

다음 `list-actions` 예제에서는 사용 가능한 작업을 나열합니다.

```
aws fis list-actions
```

출력:

```
{
    "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": {}
        }
    ]
}
```

자세한 내용은 AWS Fault Injection Simulator 사용 설명서의 작업을 참조하세요.

- 자세한 API 내용은 명령 참조ListActions의 섹션을 참조하세요. AWS CLI

### `list-experiment-templates`

다음 코드 예시에서는 `list-experiment-templates`을 사용하는 방법을 보여 줍니다.

AWS CLI

실험 템플릿을 나열하려면

다음 `list-experiment-templates` 예제에서는 AWS 계정의 실험 템플릿을 나열합니다.

```
aws fis list-experiment-templates
```

출력:

```
{
    "experimentTemplates": [
        {
            "id": "ABCDE1fgHIJkLmNop",
            "description": "myExperimentTemplate",
            "creationTime": 1616017191.124,
            "lastUpdateTime": 1616017191.124,
            "tags": {
                "key": "value"
            }
        }
    ]
}
```

자세한 내용은 AWS Fault Injection Simulator 사용 설명서의 실험 템플릿을 참조하세요.

- 자세한 API 내용은 명령 참조ListExperimentTemplates의 섹션을 참조하세요. AWS CLI

### `list-experiments`

다음 코드 예시에서는 `list-experiments`을 사용하는 방법을 보여 줍니다.

AWS CLI

실험을 나열하려면

다음 `list-experiments` 예제에서는 AWS 계정의 실험을 나열합니다.

```
aws fis list-experiments
```

출력:

```
{
    "experiments": [
        {
            "id": "ABCdeF1GHiJkLM23NO",
            "experimentTemplateId": "ABCDE1fgHIJkLmNop",
            "state": {
                "status": "running",
                "reason": "Experiment is running."
            },
```

```
            "creationTime": 1616017341.197,
            "tags": {
            "key": "value"
            }
        }
    ]
}
```

자세한 내용은 AWS Fault Injection Simulator 사용 설명서의 실험을 참조하세요.

- 자세한 API 내용은 명령 참조ListExperiments의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스의 태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 지정된 리소스의 태그를 나열합니다.

```
aws fis list-tags-for-resource \
    --resource-arn arn:aws:fis:us-west-2:123456789012:experiment/ABC12DeFGhI3jKLMNOP
```

출력:

```
{
    "tags": {
        "key1": "value1",
        "key2": "value2"
    }
}
```

자세한 내용은 AWS Fault Injection Simulator 사용 설명서의 리소스 태그 지정을 참조하세요 AWS FIS.

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### `start-experiment`

다음 코드 예시에서는 `start-experiment`을 사용하는 방법을 보여 줍니다.

AWS CLI

실험을 시작하려면

다음 `start-experiment` 예제에서는 지정된 실험을 시작합니다.

```
aws fis start-experiment \
    --experiment-template-id ABCDE1fgHIJkLmNop
```

출력:

```
{
    "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": {}
    }
}
```

자세한 내용은 AWS Fault Injection Simulator 사용 설명서의 실험 AWS FIS을 참조하세요.

- 자세한 API 내용은 명령 참조StartExperiment의 섹션을 참조하세요. AWS CLI

### stop-experiment

다음 코드 예시에서는 stop-experiment을 사용하는 방법을 보여 줍니다.

AWS CLI

실험을 중지하려면

다음 stop-experiment 예제는 지정된 실험의 실행을 중지합니다.

```
aws fis stop-experiment \
    --id ABC12DeFGhI3jKLMNOP
```

출력:

```
{
    "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": {}
    }
}
```

자세한 내용은 AWS Fault Injection Simulator 사용 설명서의 실험 AWS FIS을 참조하세요.

- 자세한 API 내용은 명령 참조StopExperiment의 섹션을 참조하세요. AWS CLI

### tag-resource

다음 코드 예시에서는 tag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 태그를 지정하려면

다음 tag-resource 예제에서는 지정된 리소스에 태그를 지정합니다.

```
aws fis tag-resource \
    --resource-arn arn:aws:fis:us-west-2:123456789012:experiment/ABC12DeFGhI3jKLMNOP
 \
    --tags key1=value1,key2=value2
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Fault Injection Simulator 사용 설명서의 리소스 태그 지정을 참조하세요 AWS FIS.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### untag-resource

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스의 태그를 해제하려면

다음 untag-resource 예제에서는 지정된 리소스에서 태그를 제거합니다.

```
aws fis untag-resource \
    --resource-arn arn:aws:fis:us-west-2:123456789012:experiment/ABC12DeFGhI3jKLMNOP
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Fault Injection Simulator 사용 설명서의 리소스 태그 지정을 참조하세요 AWS FIS.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-experiment-template

다음 코드 예시에서는 update-experiment-template을 사용하는 방법을 보여 줍니다.

AWS CLI

실험 템플릿을 업데이트하려면

다음 update-experiment-template 예제에서는 지정된 실험 템플릿의 설명을 업데이트합니다.

```
aws fis update-experiment-template \
    --id ABCDE1fgHIJkLmNop \
    ---description myExperimentTemplate
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 AWS Fault Injection Simulator 사용 설명서의 실험 템플릿 업데이트를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateExperimentTemplate의 섹션을 참조하세요. AWS CLI

# 를 사용한 Amazon GameLift 예제 AWS CLI

다음 코드 예제에서는 Amazon 와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 GameLift.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### **create-build**

다음 코드 예시에서는 `create-build`을 사용하는 방법을 보여 줍니다.

AWS CLI

Example1: S3 버킷의 파일에서 게임 빌드를 생성하려면

다음 create-build 예제에서는 사용자 지정 게임 빌드 리소스를 생성합니다. 제어하는 AWS 계정의 S3 위치에 저장된 압축 파일을 사용합니다. 이 예제에서는 Amazon에 S3 위치에 액세스할 수 있는 GameLift 권한을 부여하는 IAM 역할을 이미 생성했다고 가정합니다. 요청이 운영 체제를 지정하지 않으므로 새 빌드 리소스의 기본값은 WINDOWS_2012입니다.

```
aws gamelift create-build \
    --storage-location file://storage-loc.json \
    --name MegaFrogRaceServer.NA \
    --build-version 12345.678
```

storage-loc.json의 콘텐츠:

```
{
    "Bucket":"MegaFrogRaceServer_NA_build_files"
    "Key":"MegaFrogRaceServer_build_123.zip"
    "RoleArn":"arn:aws:iam::123456789012:role/gamelift"
}
```

출력:

```
{
    "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"
    }
}
```

Example2: 에 파일을 수동으로 업로드하기 위한 게임 빌드 리소스를 생성하려면 GameLift

다음 `create-build` 예제에서는 새 빌드 리소스를 생성합니다. 또한 Amazon S3의 위치에 게임 빌드를 수동으로 업로드할 수 있는 스토리지 GameLift 위치 및 임시 자격 증명도 가져옵니다. 빌드를 성공적으로 업로드하면 GameLift 서비스가 빌드를 검증하고 새 빌드의 상태를 업데이트합니다.

```
aws gamelift create-build \
    --name MegaFrogRaceServer.NA \
    --build-version 12345.678 \
    --operating-system AMAZON_LINUX
```

출력:

```
{
    "Build": {
        "BuildArn": "arn:aws:gamelift:us-west-2::build/build-a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111",
        "BuildId": "build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "CreationTime": 1496708916.18,
        "Name": "MegaFrogRaceServer.NA",
        "OperatingSystem": "AMAZON_LINUX",
        "SizeOnDisk": 0,
        "Status": "INITIALIZED",
        "Version": "12345.678"
    },
    "StorageLocation": {
        "Bucket": "gamelift-builds-us-west-2",
        "Key": "123456789012/build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    },
    "UploadCredentials": {
        "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
        "SessionToken": "AgoGb3JpZ2luENz...EXAMPLETOKEN=="
    }
}
```

자세한 내용은 Amazon GameLift 개발자 안내서의 에 사용자 지정 서버 빌드 업로드 GameLift를 참조하세요.

- 자세한 API 내용은 명령 참조CreateBuild의 섹션을 참조하세요. AWS CLI

### create-fleet

다음 코드 예시에서는 create-fleet을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 기본 Linux 플릿 생성

다음 create-fleet 예제에서는 사용자 지정 서버 빌드를 호스팅하도록 최소로 구성된 온디맨드 Linux 인스턴스 플릿을 생성합니다. 를 사용하여 구성을 완료할 수 있습니다update-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}]'
```

출력:

```
{
    "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"
    }
```

```
}
```

예제 2: 기본 Windows 플릿 생성

다음 `create-fleet` 예제에서는 사용자 지정 서버 빌드를 호스팅하도록 최소로 구성된 스팟 Windows 인스턴스 플릿을 생성합니다. 를 사용하여 구성을 완료할 수 있습니다`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}]'
```

출력:

```
{
    "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"
    }
}
```

예제 3: 완전히 구성된 플릿 생성

다음 create-fleet 예제에서는 사용자 지정 서버 빌드에 대한 스팟 Windows 인스턴스 플릿을 생성하며, 가장 일반적으로 사용되는 구성 설정이 제공됩니다.

```
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'
```

runtime-config.json의 콘텐츠:

```
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}]
```

출력:

```
{
    "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"
        }
    }
}
```

예제 4: Realtime Servers 플릿 생성

다음 `create-fleet` 예제에서는 Amazon 에 업로드된 실시간 구성 스크립트를 사용하여 스팟 인스턴스 플릿을 생성합니다 GameLift. 모든 Realtime 서버는 Linux 시스템에 배포됩니다. 이 예제에서는 업로드된 Realtime 스크립트에 여러 스크립트 파일이 포함되어 있고 `Init()` 함수가 라는 스크립트 파일에 있다고 가정합니다`MainScript.js`. 그림과 같이 이 파일은 런타임 구성에서 시작 스크립트로 식별됩니다.

```
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}]'
```

출력:

```
{
    "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"
    }
}
```

- 자세한 API 내용은 명령 참조CreateFleet의 섹션을 참조하세요. AWS CLI

### `create-game-session-queue`

다음 코드 예시에서는 `create-game-session-queue`을 사용하는 방법을 보여 줍니다.

AWS CLI

Example1: 순서가 지정된 게임 세션 대기열을 설정하려면

다음 `create-game-session-queue` 예제에서는 두 리전의 대상을 사용하여 새 게임 세션 대기열을 생성합니다. 또한 배치를 위해 10분을 기다린 후 게임 세션이 제한 시간을 요청하도록 대기열을 구성합니다. 지연 시간 정책이 정의되지 않았으므로 는 첫 번째 대상이 나열된 모든 게임 세션을 배치하려고 GameLift 시도합니다.

```
aws gamelift create-game-session-queue \
    --name MegaFrogRaceServer-NA \
    --destinations file://destinations.json \
    --timeout-in-seconds 600
```

`destinations.json`의 콘텐츠:

```
{
    "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" }
    ]
}
```

출력:

```
{
    "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"}
            ]
        }
    ]
}
```

Example2: 플레이어 지연 시간 정책을 사용하여 게임 세션 대기열을 설정하려면

다음 `create-game-session-queue` 예제에서는 두 개의 플레이어 지연 시간 정책을 사용하여 새 게임 세션 대기열을 생성합니다. 첫 번째 정책은 게임 세션 배치 시도의 첫 1분 동안 적용되는 100ms 지연 시간 한도를 설정합니다. 두 번째 정책은 배치 요청 시간이 3분으로 초과될 때까지 지연 시간 한도를 200ms로 높입니다.

```
aws gamelift create-game-session-queue \
    --name MegaFrogRaceServer-NA \
    --destinations file://destinations.json \
    --player-latency-policies file://latency-policies.json \
    --timeout-in-seconds 180
```

`destinations.json`의 콘텐츠:

```
{
    "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" }
    ]
}
```

`latency-policies.json`의 콘텐츠:

```
{
    "PlayerLatencyPolicies": [
        {"MaximumIndividualPlayerLatencyMilliseconds": 200},
        {"MaximumIndividualPlayerLatencyMilliseconds": 100, "PolicyDurationSeconds":
 60}
    ]
}
```

출력:

```
{
    "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"}
        ],
    }
}
```

자세한 내용은 Amazon GameLift 개발자 안내서의 대기열 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateGameSessionQueue의 섹션을 참조하세요. AWS CLI

### delete-build

다음 코드 예시에서는 delete-build을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 게임 빌드를 삭제하려면

다음 delete-build 예제에서는 Amazon GameLift 계정에서 빌드를 제거합니다. 빌드가 삭제된 후에는 새 플릿을 생성하는 데 사용할 수 없습니다. 이 작업은 실행 취소할 수 없습니다.

```
aws gamelift delete-build \
   --build-id build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeleteBuild의 섹션을 참조하세요. AWS CLI

### delete-fleet

다음 코드 예시에서는 delete-fleet을 사용하는 방법을 보여 줍니다.

AWS CLI

더 이상 사용되지 않는 플릿을 삭제하려면

다음 delete-fleet 예제에서는 인스턴스가 0으로 축소된 플릿을 제거합니다. 플릿 용량이 0보다 크면 요청이 실패하고 HTTP 400 오류가 발생합니다.

```
aws gamelift delete-fleet \
   --fleet-id fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon GameLift 개발자 안내서의 GameLift 플릿 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteFleet의 섹션을 참조하세요. AWS CLI

### delete-game-session-queue

다음 코드 예시에서는 delete-game-session-queue을 사용하는 방법을 보여 줍니다.

AWS CLI

게임 세션 대기열을 삭제하려면

다음 delete-game-session-queue 예제에서는 지정된 게임 세션 대기열을 삭제합니다.

```
aws gamelift delete-game-session-queue \
    --name MegaFrogRace-NA
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeleteGameSessionQueue의 섹션을 참조하세요. AWS CLI

### describe-build

다음 코드 예시에서는 describe-build을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 게임 빌드에 대한 정보를 얻으려면

다음 describe-build 예제에서는 게임 서버 빌드 리소스의 속성을 검색합니다.

```
aws gamelift describe-build \
    --build-id build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

출력:

```
{
    "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"
```

```
    }
}
```

자세한 내용은 Amazon GameLift 개발자 안내서의 에 사용자 지정 서버 빌드 업로드 GameLift를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeBuild의 섹션을 참조하세요. AWS CLI

### `describe-ec2-instance-limits`

다음 코드 예시에서는 `describe-ec2-instance-limits`을 사용하는 방법을 보여 줍니다.

AWS CLI

EC2 인스턴스 유형에 대한 서비스 제한을 검색하려면

다음 `describe-ec2-instance-limits` 예제에서는 현재 리전에서 지정된 인스턴스 유형에 대해 사용 중인 최대 허용 인스턴스와 현재 EC2 인스턴스를 보여줍니다. 결과는 허용되는 20개의 인스턴스 중 5개만 사용되고 있음을 나타냅니다.

```
aws gamelift describe-ec2-instance-limits \
    --ec2-instance-type m5.large
```

출력:

```
{
    "EC2InstanceLimits": [
        {
            "EC2InstanceType": ""m5.large",
            "CurrentInstances": 5,
            "InstanceLimit": 20
        }
    ]
}
```

자세한 내용은 Amazon GameLift 개발자 안내서의 컴퓨팅 리소스 선택을 참조하세요.

- API 자세한 내용은 AWS CLI 명령 참조의 DescribeEc2InstanceLimits를 참조하세요.

### `describe-fleet-attributes`

다음 코드 예시에서는 `describe-fleet-attributes`을 사용하는 방법을 보여 줍니다.

AWS CLI

Example1: 플릿 목록의 속성을 보려면

다음 describe-fleet-attributes 예제에서는 지정된 두 플릿에 대한 플릿 속성을 검색합니다. 그림과 같이 요청된 플릿은 온디맨드 인스턴스용과 스팟 인스턴스용의 동일한 빌드로 배포되며 약간의 구성 차이가 있습니다.

```
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
```

출력:

```
{
    "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"
            }
        }
    ]
}
```

Example2: 모든 플릿에 대한 속성을 요청하려면

다음은 모든 상태의 모든 플릿에 대한 플릿 속성을 describe-fleet-attributes 반환합니다. 이 예제에서는 페이지 매김 파라미터를 사용하여 한 번에 하나의 플릿을 반환하는 방법을 보여줍니다.

```
aws gamelift describe-fleet-attributes \
    --limit 1
```

출력:

```
{
    "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":
 "eyJhd3NBY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZTZmMS
}
```

출력에는 명령을 두 번째로 호출할 때 사용할 수 있는 NextToken 값이 포함됩니다. 값을 --next-token 파라미터에 전달하여 출력을 선택할 위치를 지정합니다. 다음 명령은 출력의 두 번째 결과를 반환합니다.

```
aws gamelift describe-fleet-attributes \
    --limit 1 \
    --next-
token eyJhd3NBY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZT
```

응답에 NextToken 값이 포함되지 않을 때까지 반복합니다.

자세한 내용은 Amazon GameLift 개발자 안내서의 GameLift 플릿 설정을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeFleetAttributes의 섹션을 참조하세요. AWS CLI

### describe-fleet-capacity

다음 코드 예시에서는 describe-fleet-capacity을 사용하는 방법을 보여 줍니다.

AWS CLI

플릿 목록의 용량 상태를 보려면

다음 `describe-fleet-capacity` 예제에서는 지정된 두 플릿의 현재 용량을 검색합니다.

```
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
```

출력:

```
{
    "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
            }
        }

    ]
}
```

자세한 내용은 Amazon GameLift 개발자 안내서의 GameLift 플릿 지표를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeFleetCapacity의 섹션을 참조하세요. AWS CLI

### describe-fleet-events

다음 코드 예시에서는 describe-fleet-events을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 기간 동안 이벤트를 요청하려면

다음 describe-fleet-events 예제에서는 지정된 기간 동안 발생한 모든 플릿 관련 이벤트의 세부 정보를 표시합니다.

```
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
```

출력:

```
{
    "Events": [
        {
            "EventId": "a37b6892-5d07-4d3b-8b47-80244ecf66b9",
            "ResourceId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "EventCode": "FLEET_STATE_ACTIVE",
            "Message": "Fleet fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 changed
 state to ACTIVE",
            "EventTime": 1579649342.191
        },
        {
            "EventId": "67da4ec9-92a3-4d95-886a-5d6772c24063",
            "ResourceId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "EventCode": "FLEET_STATE_ACTIVATING",
            "Message": "Fleet fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 changed
 state to ACTIVATING",
            "EventTime": 1579649321.427
        },
        {
```

```
            "EventId": "23813a46-a9e6-4a53-8847-f12e6a8381ac",
            "ResourceId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "EventCode": "FLEET_STATE_BUILDING",
            "Message": "Fleet fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 changed
 state to BUILDING",
            "EventTime": 1579649321.243
        },
        {
            "EventId": "3bf217d0-1d44-42f9-9202-433ed475d2e8",
            "ResourceId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "EventCode": "FLEET_STATE_VALIDATING",
            "Message": "Fleet fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 changed
 state to VALIDATING",
            "EventTime": 1579649197.449
        },
        {
            "EventId": "2ecd0130-5986-44eb-99a7-62df27741084",
            "ResourceId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "EventCode": "FLEET_VALIDATION_LAUNCH_PATH_NOT_FOUND",
            "Message": "Failed to find a valid path",
            "EventTime": 1569319075.839,
            "PreSignedLogUrl": "https://gamelift-event-logs-prod-
us-west-2.s3.us-west-2.amazonaws.com/logs/fleet-83422059-8329-42a2-
a4d6-c4444386a6f8/events/2ecd0130-5986-44eb-99a7-62df27741084/
FLEET_VALIDATION_LAUNCH_PATH_NOT_FOUND.txt?X-Amz-Security-
Token=IQoJb3JpZ2luX2VjEB8aCXVzLXdlc3QtMiJHMEUCIHV5K%2FLPx8h310D
%2FAvx0%2FZxsDy5XA3cJOwPdu3T0eBa%2FAiEA1yovokcZYy%2FV4CWW6l26aFyiSHO
%2Bxz%2FBMAhEHYHMQNcqkQMImP%2F%2F%2F%2F%2F%2F%2F%2F%2F%2F
%2FARAAGgw3NDEwNjE1OTIxNzEiDI8rsZtzLzlwEDQhXSrlAtl5Ae
%2Fgo6FCIzqXPbXfBOnSvFYqeDlriZarEpKqKrUt8mXQv9iqHResqCph9AKo49lwgSYTT2QoSxnrD7%2FUgv
%2BZm2pVuczvuKtUA0fcx6s0GxpjIAzdIE%2F5P%2FB7B9M%2BVZ
%2F9KF82hbJi0HTE6Y7BjKsEgFCvk4UXILhfjtan9iQl8%2F21ZTurAcJbm7Y5tuLF9SWSK3%2BEa7VXOcCK4D4O1sMj
%2FIaXoHkNvg0RVTa0hIqdvpaDQlsSBNdqTXbjHTu6fETE9Y9Ky%2BiJK5KiUG
%2F59GjCpDcvS1FqKeLUEmKT7wysGmvjMc2n%2Fr
%2F9VxQfte7w9srXwlLAQuwhiXAAyI5ICMZ5JvzjzQwTqD4CHTVKUUDwL
%2BRZzbuuqkJObZml02CkRGp%2B74RTAzLbWptVqZTIfzctiCTmWxb
%2FmKyELRYsVLrwNJ%2BGJ7%2BCrN0RC%2FjlgfLYIZyeAqjPgAu5HjgX
%2BM7jCo9M7wBTrnAXKOFQuf9dvA84SuwXOJFp17LYGjrHMKv0qC3GfbTMrZ6kzeNV9awKCpXB2Gnx9z2KvIlJdqirWV
%2F9C6%2B4jIZPME3jXmZcEHqqw5uvAVF7aeIavtUZU8pxpDIWT0YE4p3Kriy2AA7ziCRKtVfjV839InyLk8LUjsioWK
%2BYUq8%2FDTlLxqj1S%2FiO4TI0Wo7ilAo%2FKKWWF4guuNDexj8EOOynSp1yImB
%2BZf2Fua3O44W4eEXAMPLE33333&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Date=20170621T231808Z&X-Amz-SignedHeaders=host&X-Amz-Expires=900&X-Amz-
Credential=AKIAIOSFODNN7EXAMPLE%2F20170621%2Fus-west-2%2Fs3%2Faws4_request&X-Amz-
Signature=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY"
```

```
        }
    ],
    "NextToken":
 "eyJhd3NBY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZTZmMS
}
```

자세한 내용은 Amazon GameLift 개발자 안내서의 GameLift 플릿 문제 디버그를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeFleetEvents의 섹션을 참조하세요. AWS CLI

### describe-fleet-port-settings

다음 코드 예시에서는 describe-fleet-port-settings을 사용하는 방법을 보여 줍니다.

AWS CLI

플릿에 대한 인바운드 연결 권한을 보려면

다음 describe-fleet-port-settings 예제에서는 지정된 플릿에 대한 연결 설정을 검색합니다.

```
aws gamelift describe-fleet-port-settings \
    --fleet-id arn:aws:gamelift:us-west-2::fleet/fleet-a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111
```

출력:

```
{
    "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"
        }
    ]
```

```
}
```

자세한 내용은 Amazon GameLift 개발자 안내서의 GameLift 플릿 설정을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeFleetPortSettings의 섹션을 참조하세요. AWS CLI

### describe-fleet-utilization

다음 코드 예시에서는 describe-fleet-utilization을 사용하는 방법을 보여 줍니다.

AWS CLI

Example1: 플릿 목록의 사용 데이터를 보려면

다음 describe-fleet-utilization 예제에서는 지정된 플릿 하나에 대한 현재 사용 정보를 검색합니다.

```
aws gamelift describe-fleet-utilization \
    --fleet-ids arn:aws:gamelift:us-west-2::fleet/fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

출력:

```
{
    "FleetUtilization": [
        {
        "FleetId": "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "ActiveServerProcessCount": 100,
        "ActiveGameSessionCount": 62,
        "CurrentPlayerSessionCount": 329,
        "MaximumPlayerSessionCount": 1000
        }
    ]
}
```

Example2: 모든 플릿에 대한 사용 데이터를 요청하려면

다음은 모든 상태의 모든 플릿에 대한 플릿 사용 데이터를 describe-fleet-utilization 반환합니다. 이 예제에서는 페이지 매김 파라미터를 사용하여 한 번에 두 플릿의 데이터를 반환합니다.

```
aws gamelift describe-fleet-utilization \
```

```
    --limit 2
```

출력:

```
{
    "FleetUtilization": [
        {
            "FleetId": "fleet-1111aaaa-22bb-33cc-44dd-5555eeee66ff",
            "ActiveServerProcessCount": 100,
            "ActiveGameSessionCount": 13,
            "CurrentPlayerSessionCount": 98,
            "MaximumPlayerSessionCount": 1000
        },
        {
            "FleetId": "fleet-2222bbbb-33cc-44dd-55ee-6666ffff77aa",
            "ActiveServerProcessCount": 100,
            "ActiveGameSessionCount": 62,
            "CurrentPlayerSessionCount": 329,
            "MaximumPlayerSessionCount": 1000
        }
    ],
    "NextToken":
 "eyJhd3NBY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZTZmMS
}
```

명령을 두 번째로 호출하여 NextToken 값을 --next-token 파라미터에 인수로 전달하여 다음 두 결과를 확인합니다.

```
aws gamelift describe-fleet-utilization \
    --limit 2 \
    --next-
token eyJhd3NBY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZT
```

응답에 더 이상 값이 출력NextToken에 포함되지 않을 때까지 반복합니다.

자세한 내용은 Amazon GameLift 개발자 안내서의 GameLift 플릿 지표를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeFleetUtilization의 섹션을 참조하세요. AWS CLI

### describe-game-session-queues

다음 코드 예시에서는 describe-game-session-queues을 사용하는 방법을 보여 줍니다.

AWS CLI

게임 세션 대기열을 보려면

다음 `describe-game-session-queues` 예제에서는 지정된 두 대기열의 속성을 검색합니다.

```
aws gamelift describe-game-session-queues \
    --names MegaFrogRace-NA MegaFrogRace-EU
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon GameLift 개발자 안내서의 다중 리전 대기열 사용을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeGameSessionQueues의 섹션을 참조하세요. AWS CLI

### describe-runtime-configuration

다음 코드 예시에서는 describe-runtime-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

플릿에 대한 런타임 구성을 요청하려면

다음 describe-runtime-configuration 예제에서는 지정된 플릿의 현재 런타임 구성에 대한 세부 정보를 검색합니다.

```
aws gamelift describe-runtime-configuration \
    --fleet-id fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 Amazon GameLift 개발자 안내서의 플릿에서 여러 프로세스 실행을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeRuntimeConfiguration의 섹션을 참조하세요. AWS CLI

### list-builds

다음 코드 예시에서는 list-builds을 사용하는 방법을 보여 줍니다.

AWS CLI

Example1: 사용자 지정 게임 빌드 목록을 가져오려면

다음 list-builds 예제에서는 현재 리전의 모든 게임 서버 빌드에 대한 속성을 검색합니다. 샘플 요청은 페이지 매김 파라미터 Limit 및 를 사용하여 순차적 세트로 결과를 검색NextToken하는 방법을 보여줍니다. 첫 번째 명령은 처음 두 빌드를 검색합니다. 사용 가능한 결과가 두 개 이상이기 때문에 응답에는 더 많은 결과를 사용할 수 있음을 NextToken 나타내는 가 포함됩니다.

```
aws gamelift list-builds \
    --limit 2
```

출력:

```
{
    "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":
 "eyJhd3NBY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZTZmMS
}
```

그런 다음 다음과 같이 `--next-token` 파라미터를 사용하여 명령을 다시 호출하여 다음 두 빌드를 볼 수 있습니다.

```
aws gamelift list-builds \
    --limit 2
    --next-
token eyJhd3NBY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZT
```

응답에 `NextToken` 값이 포함되지 않을 때까지 반복합니다.

Example2: 실패 상태의 사용자 지정 게임 빌드 목록을 가져오려면

다음 `list-builds` 예제는 현재 상태가 인 현재 리전의 모든 게임 서버 빌드에 대한 속성을 검색합니다FAILED.

```
aws gamelift list-builds \
    --status FAILED
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListBuilds의 섹션을 참조하세요. AWS CLI

### `list-fleets`

다음 코드 예시에서는 `list-fleets`을 사용하는 방법을 보여 줍니다.

AWS CLI

Example1: 리전의 모든 플릿 목록을 가져오려면

다음 `list-fleets` 예제에서는 현재 리전IDs의 모든 플릿의 플릿을 표시합니다. 이 예제에서는 페이지 매김 파라미터를 사용하여 IDs 한 번에 두 개의 플릿을 검색합니다. 응답에는 검색할 결과가 더 많음을 나타내는 `next-token` 속성이 포함됩니다.

```
aws gamelift list-fleets \
    --limit 2
```

출력:

```
{
    "FleetIds": [
        "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
    ],
    "NextToken":
 "eyJhd3NBY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC01NWYxZTZmMS
}
```

다음 두 가지 결과를 얻기 위해 여기에 표시된 것처럼 다음 명령의 이전 응답에서 NextToken 값을 전달할 수 있습니다.

```
aws gamelift list-fleets \
    --limit 2 \
    --next-
token eyJhd3NBY2NvdW50SWQiOnsicyI6IjMwMjc3NjAxNjM5OCJ9LCJidWlsZElkIjp7InMiOiJidWlsZC00NDRlZj
```

Example2: 특정 빌드 또는 스크립트가 있는 리전의 모든 플릿 목록을 가져오려면

다음 list-builds 예제에서는 지정된 게임 빌드와 함께 배포된 플릿IDs의 를 검색합니다. Realtime Servers를 사용하는 경우 빌드 ID 대신 스크립트 ID를 제공할 수 있습니다. 이 예제에서는 제한 파라미터를 지정하지 않으므로 결과에 최대 16개의 플릿이 포함될 수 있습니다IDs.

```
aws gamelift list-fleets \
    --build-id build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

출력:

```
{
    "FleetIds": [
        "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
        "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
        "fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE44444"
    ]
}
```

- 자세한 API 내용은 명령 참조ListFleets의 섹션을 참조하세요. AWS CLI

### request-upload-credentials

다음 코드 예시에서는 request-upload-credentials을 사용하는 방법을 보여 줍니다.

AWS CLI

빌드 업로드를 위한 액세스 자격 증명을 새로 고치려면

다음 create-build 예제에서는 Amazon S3 위치에 GameLift 빌드 파일을 업로드하기 위한 새롭고 유효한 액세스 자격 증명을 가져옵니다. 자격 증명의 수명은 제한적입니다. 원래 CreateBuild 요청에 대한 응답에서 빌드 ID를 가져옵니다.

```
aws gamelift request-upload-credentials \
    --build-id build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

출력:

```
{
    "StorageLocation": {
        "Bucket": "gamelift-builds-us-west-2",
        "Key": "123456789012/build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    },
    "UploadCredentials": {
        "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
        "SessionToken": "AgoGb3JpZ2luENz...EXAMPLETOKEN=="
    }
}
```

자세한 내용은 Amazon GameLift 개발자 안내서의 에 사용자 지정 서버 빌드 업로드 GameLift를 참조하세요.

- 자세한 API 내용은 명령 참조RequestUploadCredentials의 섹션을 참조하세요. AWS CLI

### `start-fleet-actions`

다음 코드 예시에서는 `start-fleet-actions`을 사용하는 방법을 보여 줍니다.

AWS CLI

플릿 자동 조정 활동을 다시 시작하려면

다음 `start-fleet-actions` 예제에서는 지정된 플릿에 정의되었지만 ``stop-fleet-actions``를 호출하여 중지된 모든 조정 정책의 사용을 재개합니다. 시작 후 조정 정책은 즉시 해당 지표를 추적하기 시작합니다.

```
aws gamelift start-fleet-actions \
    --fleet-id fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --actions AUTO_SCALING
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조StartFleetActions의 섹션을 참조하세요. AWS CLI

### stop-fleet-actions

다음 코드 예시에서는 stop-fleet-actions을 사용하는 방법을 보여 줍니다.

AWS CLI

플릿의 자동 조정 활동을 중지하려면

다음 stop-fleet-actions 예제에서는 지정된 플릿에 정의된 모든 조정 정책의 사용을 중지합니다. 정책이 일시 중지된 후 수동으로 조정하지 않는 한 플릿 용량은 동일한 활성 인스턴스 수로 유지됩니다.

```
aws gamelift start-fleet-actions \
    --fleet-id fleet-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --actions AUTO_SCALING
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조StopFleetActions의 섹션을 참조하세요. AWS CLI

### update-build

다음 코드 예시에서는 update-build을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 게임 빌드를 업데이트하려면

다음 update-build 예제에서는 지정된 빌드 리소스와 연결된 이름 및 버전 정보를 변경합니다. 반환된 빌드 객체는 변경 사항이 성공적으로 이루어졌는지 확인합니다.

```
aws gamelift update-build \
    --build-id build-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --name MegaFrogRaceServer.NA.east \
    --build-version 12345.east
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon GameLift 개발자 안내서의 빌드 파일 업데이트를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateBuild의 섹션을 참조하세요. AWS CLI

### update-game-session-queue

다음 코드 예시에서는 update-game-session-queue을 사용하는 방법을 보여 줍니다.

AWS CLI

게임 세션 대기열 구성을 업데이트하려면

다음 update-game-session-queue 예제에서는 새 대상을 추가하고 기존 게임 세션 대기열에 대한 플레이어 지연 시간 정책을 업데이트합니다.

```
aws gamelift update-game-session-queue \
    --name MegaFrogRace-NA \
    --destinations file://destinations.json \
    --player-latency-policies file://latency-policies.json
```

destinations.json의 콘텐츠:

```
{
    "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"}
```

```
    ]
}
```

`latency-policies.json`의 콘텐츠:

```
{
    "PlayerLatencyPolicies": [
        {"MaximumIndividualPlayerLatencyMilliseconds": 200},
        {"MaximumIndividualPlayerLatencyMilliseconds": 150, "PolicyDurationSeconds":
 120},
        {"MaximumIndividualPlayerLatencyMilliseconds": 100, "PolicyDurationSeconds":
 120}
    ]
}
```

출력:

```
{
    "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}
        ]
    }
}
```

자세한 내용은 Amazon GameLift 개발자 안내서의 다중 리전 대기열 사용을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateGameSessionQueue의 섹션을 참조하세요. AWS CLI

### upload-build

다음 코드 예시에서는 upload-build을 사용하는 방법을 보여 줍니다.

AWS CLI

Example1: Linux 게임 서버 빌드 업로드

다음 upload-build 예제에서는 Linux 게임 서버 빌드 파일을 파일 디렉터리에서 GameLift 서비스로 업로드하고 빌드 리소스를 생성합니다.

```
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
```

출력:

```
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
```

Example2: Windows 게임 서버 빌드 업로드

다음 upload-build 예제에서는 디렉터리에서 GameLift 서비스로 Windows 게임 서버 빌드 파일을 업로드하고 빌드 레코드를 생성합니다.

```
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
```

출력:

```
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
```

자세한 내용은 Amazon GameLift 개발자 안내서의 에 사용자 지정 서버 빌드 업로드 GameLift를 참조하세요.

- 자세한 API 내용은 명령 참조UploadBuild의 섹션을 참조하세요. AWS CLI

## 를 사용한 Global Accelerator 예제 AWS CLI

다음 코드 예제에서는 Global Accelerator AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `add-custom-routing-endpoints`

다음 코드 예시에서는 `add-custom-routing-endpoints`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 라우팅 액셀러레이터의 엔드포인트 그룹에 VPC 서브넷 엔드포인트를 추가하려면

다음 add-custom-routing-endpoints 예제에서는 사용자 지정 라우팅 액셀러레이터의 엔드포인트 그룹에 VPC 서브넷 엔드포인트를 추가합니다.

```
aws globalaccelerator add-custom-routing-endpoints \
    --endpoint-group-
arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/0123vxyz/endpoint-group/4321abcd \
    --endpoint-configurations "EndpointId=subnet-1234567890abcdef0"
```

출력:

```
{
    "EndpointDescriptions": [
        {
            "EndpointId": "subnet-1234567890abcdef0"
        }
    ],

 "EndpointGroupArn":"arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh/listener/0123vxyz/endpoint-group/4321abcd"
}
```

자세한 내용은 VPC AWS Global Accelerator 개발자 안내서의 Global Accelerator에서 사용자 지정 라우팅 액셀러레이터의 서브넷 엔드포인트를 참조하세요. AWS

- 자세한 API 내용은 명령 참조AddCustomRoutingEndpoints의 섹션을 참조하세요. AWS CLI

### advertise-byoip-cidr

다음 코드 예시에서는 advertise-byoip-cidr을 사용하는 방법을 보여 줍니다.

AWS CLI

주소 범위를 광고하려면

다음 advertise-byoip-cidr 예제에서는 AWS 리소스와 함께 사용하도록 프로비저닝한 주소 범위를 광고 AWS 하라는 요청을 보여줍니다.

```
aws globalaccelerator advertise-byoip-cidr \
    --cidr 198.51.100.0/24
```

출력:

```
{
    "ByoipCidr": {
        "Cidr": "198.51.100.0/24",
        "State": "PENDING_ADVERTISING"
    }
}
```

자세한 내용은 AWS Global Accelerator 개발자 안내서의 Global Accelerator에서 고유 IP 주소 가져오기를 참조하세요. AWS

- 자세한 API 내용은 명령 참조AdvertiseByoipCidr의 섹션을 참조하세요. AWS CLI

### `allow-custom-routing-traffic`

다음 코드 예시에서는 `allow-custom-routing-traffic`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 라우팅 액셀러레이터에 대해 VPC 서브넷의 특정 Amazon EC2 인스턴스 대상으로 트래픽을 허용하려면

다음 `allow-custom-routing-traffic` 예제에서는 사용자 지정 라우팅 액셀러레이터의 VPC 서브넷 엔드포인트에 대한 특정 Amazon EC2 인스턴스(대상) IP 주소 및 포트로 트래픽이 허용되도록 지정합니다.

```
aws globalaccelerator allow-custom-routing-traffic \
    --endpoint-group-
arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/0123vxyz/endpoint-group/ab88888example \
    --endpoint-id subnet-abcd123example \
    --destination-addresses "172.31.200.6" "172.31.200.7" \
    --destination-ports 80 81
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 VPC AWS Global Accelerator 개발자 안내서의 Global Accelerator에서 사용자 지정 라우팅 액셀러레이터용 서브넷 엔드포인트를 참조하세요. AWS

- 자세한 API 내용은 명령 참조AllowCustomRoutingTraffic의 섹션을 참조하세요. AWS CLI

### create-accelerator

다음 코드 예시에서는 create-accelerator을 사용하는 방법을 보여 줍니다.

AWS CLI

액셀러레이터를 생성하려면

다음 create-accelerator 예제에서는 두 개의 BYOIP 정적 IP 주소가 있는 두 개의 태그가 있는 액셀러레이터를 생성합니다. 액셀러레이터를 생성하거나 업데이트하려면 US-West-2 (Oregon) 리전을 지정해야 합니다.

```
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
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 AWS Global Accelerator 개발자 안내서의 Global Accelerator에서 AWS Accelerator를 참조하세요.

- 자세한 API 내용은 명령 참조CreateAccelerator의 섹션을 참조하세요. AWS CLI

### `create-custom-routing-accelerator`

다음 코드 예시에서는 `create-custom-routing-accelerator`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 라우팅 액셀러레이터를 생성하려면

다음 `create-custom-routing-accelerator` 예제에서는 태그 `Name` 및 를 사용하여 사용자 지정 라우팅 액셀러레이터를 생성합니다`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
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 AWS Global Accelerator 개발자 안내서의 Global Accelerator에서 사용자 지정 라우팅 액셀러레이터를 참조하세요. AWS

- 자세한 API 내용은 명령 참조CreateCustomRoutingAccelerator의 섹션을 참조하세요. AWS CLI

### create-custom-routing-endpoint-group

다음 코드 예시에서는 create-custom-routing-endpoint-group을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 라우팅 액셀러레이터에 대한 엔드포인트 그룹을 생성하려면

다음 create-custom-routing-endpoint-group 예제에서는 사용자 지정 라우팅 액셀러레이터에 대한 엔드포인트 그룹을 생성합니다.

```
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"
```

출력:

```
{
    "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": []
    }
}
```

자세한 내용은 AWS Global Accelerator 개발자 안내서의 Global Accelerator에서 사용자 지정 라우팅 액셀러레이터용 엔드포인트 그룹을 참조하세요. AWS

- 자세한 API 내용은 명령 참조CreateCustomRoutingEndpointGroup의 섹션을 참조하세요. AWS CLI

### `create-custom-routing-listener`

다음 코드 예시에서는 `create-custom-routing-listener`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 라우팅 액셀러레이터에 대한 리스너를 생성하려면

다음 `create-custom-routing-listener` 예제에서는 사용자 지정 라우팅 액셀러레이터의 포트 범위가 5000~10000인 리스너를 생성합니다.

```
aws globalaccelerator create-custom-routing-listener \
    --accelerator-arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh \
    --port-ranges FromPort=5000,ToPort=10000
```

출력:

```
{
    "Listener": {
        "PortRange": [
            "FromPort": 5000,
            "ToPort": 10000
        ],
        "ListenerArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh/listener/0123vxyz"
    }
```

```
}
```

자세한 내용은 AWS Global Accelerator 개발자 안내서의 Global Accelerator에서 사용자 지정 라우팅 액셀러레이터용 리스너를 참조하세요. AWS

- 자세한 API 내용은 명령 참조CreateCustomRoutingListener의 섹션을 참조하세요. AWS CLI

### create-endpoint-group

다음 코드 예시에서는 create-endpoint-group을 사용하는 방법을 보여 줍니다.

AWS CLI

엔드포인트 그룹을 생성하려면

다음 create-endpoint-group 예제에서는 엔드포인트가 하나 있는 엔드포인트 그룹을 생성합니다.

```
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
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS Global Accelerator 개발자 안내서의 Global Accelerator의 엔드포인트 그룹을 참조하세요. AWS

- 자세한 API 내용은 명령 참조CreateEndpointGroup의 섹션을 참조하세요. AWS CLI

### `create-listener`

다음 코드 예시에서는 `create-listener`을 사용하는 방법을 보여 줍니다.

AWS CLI

리스너를 생성하려면

다음 `create-listener` 예제에서는 두 개의 포트가 있는 리스너를 생성합니다.

```
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
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS Global Accelerator 개발자 안내서의 Global Accelerator의 리스너를 참조하세요. AWS

- 자세한 API 내용은 명령 참조CreateListener의 섹션을 참조하세요. AWS CLI

### deny-custom-routing-traffic

다음 코드 예시에서는 deny-custom-routing-traffic을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 라우팅 액셀러레이터에서 트래픽을 수신할 수 없는 대상 주소를 지정하려면

다음 deny-custom-routing-traffic 예제에서는 사용자 지정 라우팅 액셀러레이터에 대한 트래픽을 수신할 수 없는 서브넷 엔드포인트의 대상 주소 또는 주소를 지정합니다. 둘 이상의 대상 주소를 지정하려면 주소를 공백으로 구분합니다. 성공적인 deny-custom-routing-traffic 호출에 대한 응답이 없습니다.

```
aws globalaccelerator deny-custom-routing-traffic \
    --endpoint-group-
arn "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/0123vxyz/endpoint-group/ab88888example" \
    --endpoint-id "subnet-abcd123example" \
    --destination-addresses "198.51.100.52"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 VPC AWS Global Accelerator 개발자 안내서의 Global Accelerator에서 사용자 지정 라우팅 액셀러레이터의 서브넷 엔드포인트를 참조하세요. AWS

- 자세한 API 내용은 명령 참조DenyCustomRoutingTraffic의 섹션을 참조하세요. AWS CLI

### deprovision-byoip-cidr

다음 코드 예시에서는 deprovision-byoip-cidr을 사용하는 방법을 보여 줍니다.

AWS CLI

주소 범위를 디프로비저닝하려면

다음 deprovision-byoip-cidr 예제에서는 AWS 리소스에 사용하도록 프로비저닝한 지정된 주소 범위를 릴리스합니다.

```
aws globalaccelerator deprovision-byoip-cidr \
    --cidr "198.51.100.0/24"
```

출력:

```
{
    "ByoipCidr": {
        "Cidr": "198.51.100.0/24",
        "State": "PENDING_DEPROVISIONING"
    }
}
```

자세한 내용은 AWS Global Accelerator 개발자 안내서의 Global Accelerator에서 고유 IP 주소 가져오기를 참조하세요. AWS

- 자세한 API 내용은 명령 참조DeprovisionByoipCidr의 섹션을 참조하세요. AWS CLI

### describe-accelerator-attributes

다음 코드 예시에서는 describe-accelerator-attributes을 사용하는 방법을 보여 줍니다.

AWS CLI

액셀러레이터의 속성을 설명하려면

다음 describe-accelerator-attributes 예제에서는 액셀러레이터의 속성 세부 정보를 검색합니다.

```
aws globalaccelerator describe-accelerator-attributes \
    --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh
```

출력:

```
{
    "AcceleratorAttributes": {
        "FlowLogsEnabled": true
        "FlowLogsS3Bucket": flowlogs-abc
        "FlowLogsS3Prefix": bucketprefix-abc
    }
}
```

자세한 내용은 AWS Global Accelerator 개발자 안내서의 Global Accelerator의 AWS Accelerator를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeAcceleratorAttributes의 섹션을 참조하세요. AWS CLI

### describe-accelerator

다음 코드 예시에서는 describe-accelerator을 사용하는 방법을 보여 줍니다.

AWS CLI

액셀러레이터를 설명하려면

다음 describe-accelerator 예제에서는 지정된 액셀러레이터에 대한 세부 정보를 검색합니다.

```
aws globalaccelerator describe-accelerator \
    --accelerator-arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh
```

출력:

```
{
    "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
    }
```

```
}
```

자세한 내용은 AWS Global Accelerator 개발자 안내서의 Global Accelerator의 AWS Accelerator를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeAccelerator의 섹션을 참조하세요. AWS CLI

### `describe-custom-routing-accelerator-attributes`

다음 코드 예시에서는 `describe-custom-routing-accelerator-attributes`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 라우팅 액셀러레이터의 속성을 설명하려면

다음 `describe-custom-routing-accelerator-attributes` 예제에서는 사용자 지정 라우팅 액셀러레이터의 속성을 설명합니다.

```
aws globalaccelerator describe-custom-routing-accelerator-attributes \
   --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh
```

출력:

```
{
    "AcceleratorAttributes": {
    "FlowLogsEnabled": false
    }
}
```

자세한 내용은 AWS Global Accelerator 개발자 안내서의 Global Accelerator에서 사용자 지정 라우팅 액셀러레이터를 참조하세요. AWS

- 자세한 API 내용은 명령 참조DescribeCustomRoutingAcceleratorAttributes의 섹션을 참조하세요. AWS CLI

### `describe-custom-routing-accelerator`

다음 코드 예시에서는 `describe-custom-routing-accelerator`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 라우팅 액셀러레이터를 설명하려면

다음 `describe-custom-routing-accelerator` 예제에서는 지정된 사용자 지정 라우팅 액셀러레이터에 대한 세부 정보를 검색합니다.

```
aws globalaccelerator describe-custom-routing-accelerator \
    --accelerator-arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 AWS Global Accelerator 개발자 안내서의 Global Accelerator에서 사용자 지정 라우팅 액셀러레이터를 참조하세요. AWS

- 자세한 API 내용은 명령 참조DescribeCustomRoutingAccelerator의 섹션을 참조하세요. AWS CLI

### describe-custom-routing-endpoint-group

다음 코드 예시에서는 describe-custom-routing-endpoint-group을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 라우팅 액셀러레이터의 엔드포인트 그룹을 설명하려면

다음 describe-custom-routing-endpoint-group 예제에서는 사용자 지정 라우팅 액셀러레이터의 엔드포인트 그룹에 대해 설명합니다.

```
aws globalaccelerator describe-custom-routing-endpoint-group \
    --endpoint-group-
arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/6789vxyz/endpoint-group/ab88888example
```

출력:

```
{
    "EndpointGroup": {
        "EndpointGroupArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/6789vxyz/endpoint-group/ab88888example",
        "EndpointGroupRegion": "us-east-2",
        "DestinationDescriptions": [
            {
                "FromPort": 5000,
                "ToPort": 10000,
                "Protocols": [
                    "UDP"
                ]
            }
        ],
        "EndpointDescriptions": [
            {
                "EndpointId": "subnet-1234567890abcdef0"
            }
        ]
    }
}
```

자세한 내용은 AWS Global Accelerator 개발자 안내서의 Global Accelerator에서 사용자 지정 라우팅 액셀러레이터의 엔드포인트 그룹을 참조하세요. AWS

- 자세한 API 내용은 명령 참조DescribeCustomRoutingEndpointGroup의 섹션을 참조하세요. AWS CLI

### `describe-custom-routing-listener`

다음 코드 예시에서는 `describe-custom-routing-listener`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 라우팅 액셀러레이터의 리스너를 설명하려면

다음 `describe-custom-routing-listener` 예제에서는 사용자 지정 라우팅 액셀러레이터의 리스너에 대해 설명합니다.

```
aws globalaccelerator describe-custom-routing-listener \
    --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh/listener/abcdef1234
```

출력:

```
{
    "Listener": {
        "PortRanges": [
            "FromPort": 5000,
            "ToPort": 10000
        ],
        "ListenerArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh/listener/abcdef1234"
    }
}
```

자세한 내용은 AWS Global Accelerator 개발자 안내서의 Global Accelerator에서 사용자 지정 라우팅 액셀러레이터용 리스너를 참조하세요. AWS

- 자세한 API 내용은 명령 참조DescribeCustomRoutingListener의 섹션을 참조하세요. AWS CLI

### describe-endpoint-group

다음 코드 예시에서는 describe-endpoint-group을 사용하는 방법을 보여 줍니다.

AWS CLI

엔드포인트 그룹을 설명하려면

다음 describe-endpoint-group 예제에서는 Amazon EC2 인스턴스, 및 엔드포인트가 있는 엔드포인트 그룹에 대한 세부 정보를 검색합니다ALBNLB.

```
aws globalaccelerator describe-endpoint-group \
    --endpoint-group-
arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/6789vxyz-vxyz-6789-vxyz-6789lmnopqrs/endpoint-group/
ab88888example
```

출력:

```
{
    "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"
```

```
    }
}
```

자세한 내용은 AWS Global Accelerator 개발자 안내서의 Global Accelerator의 엔드포인트 그룹을 참조하세요. AWS

- 자세한 API 내용은 명령 참조DescribeEndpointGroup의 섹션을 참조하세요. AWS CLI

### `describe-listener`

다음 코드 예시에서는 `describe-listener`을 사용하는 방법을 보여 줍니다.

AWS CLI

리스너를 설명하려면

다음 `describe-listener` 예제에서는 리스너를 설명합니다.

```
aws globalaccelerator describe-listener \
    --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh/listener/abcdef1234
```

출력:

```
{
    "Listener": {
        "ListenerArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh/listener/abcdef1234",
        "PortRanges": [
            {
                "FromPort": 80,
                "ToPort": 80
            }
        ],
        "Protocol": "TCP",
        "ClientAffinity": "NONE"
    }
}
```

자세한 내용은 AWS Global Accelerator 개발자 안내서의 Global Accelerator의 리스너를 참조하세요. AWS

- 자세한 API 내용은 명령 참조DescribeListener의 섹션을 참조하세요. AWS CLI

### `list-accelerators`

다음 코드 예시에서는 `list-accelerators`을 사용하는 방법을 보여 줍니다.

AWS CLI

액셀러레이터를 나열하려면

다음 `list-accelerators` 예제에서는 AWS 계정의 액셀러레이터를 나열합니다. 이 계정에는 두 개의 액셀러레이터가 있습니다.

```
aws globalaccelerator list-accelerators
```

출력:

```
{
    "Accelerators": [
        {
            "AcceleratorArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/5555abcd-abcd-5555-
abcd-5555EXAMPLE1",
            "Name": "TestAccelerator",
            "IpAddressType": "IPV4",
            "Enabled": true,
            "IpSets": [
                {
                    "IpFamily": "IPv4",
                    "IpAddresses": [
                        "192.0.2.250",
                        "198.51.100.52"
                    ]
                }
            ],
            "DnsName": "5a5a5a5a5a5a5a5a.awsglobalaccelerator.com",
            "Status": "DEPLOYED",
            "CreatedTime": 1552424416.0,
            "LastModifiedTime": 1569375641.0
        },
        {
```

```
            "AcceleratorArn":
 "arn:aws:globalaccelerator::888888888888:accelerator/8888abcd-abcd-8888-
abcd-8888EXAMPLE2",
            "Name": "ExampleAccelerator",
            "IpAddressType": "IPV4",
            "Enabled": true,
            "IpSets": [
                {
                    "IpFamily": "IPv4",
                    "IpAddresses": [
                        "192.0.2.100",
                        "198.51.100.10"
                    ]
                }
            ],
            "DnsName": "6a6a6a6a6a6a6a.awsglobalaccelerator.com",
            "Status": "DEPLOYED",
            "CreatedTime": 1575585564.0,
            "LastModifiedTime": 1579809243.0
        },
    ]
}
```

자세한 내용은 AWS Global Accelerator 개발자 안내서의 Global Accelerator의 AWS Accelerator를 참조하세요.

- 자세한 API 내용은 명령 참조ListAccelerators의 섹션을 참조하세요. AWS CLI

### `list-byoip-cidr`

다음 코드 예시에서는 `list-byoip-cidr`을 사용하는 방법을 보여 줍니다.

AWS CLI

주소 범위를 나열하려면

다음 `list-byoip-cidr` 예제에서는 Global Accelerator와 함께 사용하도록 프로비저닝한 고유한 IP 주소 가져오기(BYOIP) 주소 범위를 나열합니다.

```
aws globalaccelerator list-byoip-cidrs
```

출력:

```
{
    "ByoipCidrs": [
        {
            "Cidr": "198.51.100.0/24",
            "State": "READY"
        }
        {
            "Cidr": "203.0.113.25/24",
            "State": "READY"
        }
    ]
}
```

자세한 내용은 AWS Global Accelerator 개발자 안내서의 Global Accelerator에서 고유 IP 주소 가져오기를 참조하세요. AWS

- 자세한 API 내용은 명령 참조ListByoipCidr의 섹션을 참조하세요. AWS CLI

### `list-custom-routing-accelerators`

다음 코드 예시에서는 `list-custom-routing-accelerators`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 라우팅 액셀러레이터를 나열하려면

다음 `list-custom-routing-accelerators` 예제에서는 AWS 계정의 사용자 지정 라우팅 액셀러레이터를 나열합니다.

```
aws globalaccelerator list-custom-routing-accelerators
```

출력:

```
{
    "Accelerators": [
        {
            "AcceleratorArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/5555abcd-abcd-5555-
abcd-5555EXAMPLE1",
            "Name": "TestCustomRoutingAccelerator",
            "IpAddressType": "IPV4",
```

```
            "Enabled": true,
            "IpSets": [
                {
                    "IpFamily": "IPv4",
                    "IpAddresses": [
                        "192.0.2.250",
                        "198.51.100.52"
                    ]
                }
            ],
            "DnsName": "5a5a5a5a5a5a5a5a.awsglobalaccelerator.com",
            "Status": "DEPLOYED",
            "CreatedTime": 1552424416.0,
            "LastModifiedTime": 1569375641.0
        },
        {
            "AcceleratorArn":
 "arn:aws:globalaccelerator::888888888888:accelerator/8888abcd-abcd-8888-
abcd-8888EXAMPLE2",
            "Name": "ExampleCustomRoutingAccelerator",
            "IpAddressType": "IPV4",
            "Enabled": true,
            "IpSets": [
                {
                    "IpFamily": "IPv4",
                    "IpAddresses": [
                        "192.0.2.100",
                        "198.51.100.10"
                    ]
                }
            ],
            "DnsName": "6a6a6a6a6a6a6a6a.awsglobalaccelerator.com",
            "Status": "DEPLOYED",
            "CreatedTime": 1575585564.0,
            "LastModifiedTime": 1579809243.0
        },
    ]
}
```

자세한 내용은 AWS Global Accelerator 개발자 안내서의 Global Accelerator에서 사용자 지정 라우팅 액셀러레이터를 참조하세요.

- 자세한 API 내용은 명령 참조의 ListCustomRoutingAccelerators 섹션을 참조하세요. AWS CLI

### list-custom-routing-endpoint-groups

다음 코드 예시에서는 list-custom-routing-endpoint-groups을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 라우팅 액셀러레이터에서 리스너의 엔드포인트 그룹을 나열하려면

다음 list-custom-routing-endpoint-groups 예제에서는 사용자 지정 라우팅 액셀러레이터의 리스너에 대한 엔드포인트 그룹을 나열합니다.

```
aws globalaccelerator list-custom-routing-endpoint-groups \
    --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh/listener/abcdef1234
```

출력:

```
{
    "EndpointGroups": [
        {
            "EndpointGroupArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/abcdef1234/endpoint-group/ab88888example",
            "EndpointGroupRegion": "eu-central-1",
            "DestinationDescriptions": [
                {
                    "FromPort": 80,
                    "ToPort": 80,
                    "Protocols": [
                        "TCP",
                        "UDP"
                    ]
                }
            ]
            "EndpointDescriptions": [
                {
                    "EndpointId": "subnet-abcd123example"
                }
            ]
        }
    ]
}
```

자세한 내용은 AWS Global Accelerator 개발자 안내서의 Global Accelerator에서 사용자 지정 라우팅 액셀러레이터의 엔드포인트 그룹을 참조하세요. AWS

- 자세한 API 내용은 명령 참조ListCustomRoutingEndpointGroups의 섹션을 참조하세요. AWS CLI

### `list-custom-routing-listeners`

다음 코드 예시에서는 `list-custom-routing-listeners`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 라우팅 액셀러레이터의 리스너를 나열하려면

다음 `list-custom-routing-listeners` 예제에서는 사용자 지정 라우팅 액셀러레이터의 리스너를 나열합니다.

```
aws globalaccelerator list-custom-routing-listeners \
    --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh
```

출력:

```
{
    "Listeners": [
        {
            "ListenerArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh/listener/abcdef1234",
            "PortRanges": [
                {
                    "FromPort": 5000,
                    "ToPort": 10000
                }
            ],
            "Protocol": "TCP"
        }
    ]
}
```

자세한 내용은 AWS Global Accelerator 개발자 안내서의 Global Accelerator에서 사용자 지정 라우팅 액셀러레이터용 리스너를 참조하세요. AWS

- 자세한 API 내용은 명령 참조ListCustomRoutingListeners의 섹션을 참조하세요. AWS CLI

### list-custom-routing-port-mappings-by-destination

다음 코드 예시에서는 list-custom-routing-port-mappings-by-destination을 사용하는 방법을 보여 줍니다.

AWS CLI

특정 사용자 지정 라우팅 액셀러레이터 대상의 포트 매핑을 나열하려면

다음 list-custom-routing-port-mappings-by-destination 예제에서는 사용자 지정 라우팅 액셀러레이터의 특정 대상 EC2 서버(대상 주소)에 대한 포트 매핑을 제공합니다.

```
aws globalaccelerator list-custom-routing-port-mappings-by-destination \
    --endpoint-id subnet-abcd123example \
    --destination-address 198.51.100.52
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS Global Accelerator 개발자 안내서의 Global Accelerator에서 사용자 지정 라우팅 액셀러레이터 작동 방식을 참조하세요. AWS

- 자세한 API 내용은 명령 참조ListCustomRoutingPortMappingsByDestination의 섹션을 참조하세요. AWS CLI

### `list-custom-routing-port-mappings`

다음 코드 예시에서는 `list-custom-routing-port-mappings`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 라우팅 액셀러레이터에서 포트 매핑을 나열하려면

다음 `list-custom-routing-port-mappings` 예제에서는 사용자 지정 라우팅 액셀러레이터의 포트 매핑의 일부 목록을 제공합니다.

```
aws globalaccelerator list-custom-routing-port-mappings \
    --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS Global Accelerator 개발자 안내서의 Global Accelerator에서 사용자 지정 라우팅 액셀러레이터 작동 방식을 참조하세요. AWS

- 자세한 API 내용은 명령 참조ListCustomRoutingPortMappings의 섹션을 참조하세요. AWS CLI

### `list-endpoint-groups`

다음 코드 예시에서는 `list-endpoint-groups`을 사용하는 방법을 보여 줍니다.

AWS CLI

엔드포인트 그룹을 나열하려면

다음 `list-endpoint-groups` 예제에서는 리스너의 엔드포인트 그룹을 나열합니다. 이 리스너에는 두 개의 엔드포인트 그룹이 있습니다.

```
aws globalaccelerator --region us-west-2 list-endpoint-groups \
```

```
    --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh/listener/abcdef1234
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 AWS Global Accelerator 개발자 안내서의 Global Accelerator의 엔드포인트 그룹을 참조하세요. AWS

- 자세한 API 내용은 명령 참조ListEndpointGroups의 섹션을 참조하세요. AWS CLI

### `list-listeners`

다음 코드 예시에서는 `list-listeners`을 사용하는 방법을 보여 줍니다.

AWS CLI

리스너를 나열하려면

다음 `list-listeners` 예제에서는 액셀러레이터의 리스너를 나열합니다.

```
aws globalaccelerator list-listeners \
    --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh
```

출력:

```
{
    "Listeners": [
        {
            "ListenerArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh/listener/abcdef1234",
            "PortRanges": [
                {
                    "FromPort": 80,
                    "ToPort": 80
                }
            ],
            "Protocol": "TCP",
            "ClientAffinity": "NONE"
        }
    ]
}
```

자세한 내용은 AWS Global Accelerator 개발자 안내서의 Global Accelerator의 리스너를 참조하세요. AWS

- 자세한 API 내용은 명령 참조ListListeners의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

액셀러레이터의 태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 특정 액셀러레이터의 태그를 나열합니다.

```
aws globalaccelerator list-tags-for-resource \
    --accelerator-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh
```

출력:

```
{
    "Tags": [
        {
            "Key": "Project",
            "Value": "A123456"
        }
    ]
}
```

자세한 내용은 AWS Global Accelerator 개발자 안내서의 Global Accelerator에서 태그 지정을 참조하세요. AWS

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### `provision-byoip-cidr`

다음 코드 예시에서는 `provision-byoip-cidr`을 사용하는 방법을 보여 줍니다.

AWS CLI

주소 범위를 프로비저닝하려면

다음 `provision-byoip-cidr` 예제에서는 AWS 리소스에 사용할 지정된 주소 범위를 프로비저닝합니다.

```
aws globalaccelerator provision-byoip-cidr \
    --cidr 192.0.2.250/24 \
    --cidr-authorization-context Message="$text_message",Signature="$signed_message"
```

출력:

```
{
```

```
    "ByoipCidr": {
        "Cidr": "192.0.2.250/24",
        "State": "PENDING_PROVISIONING"
    }
}
```

자세한 내용은 AWS Global Accelerator 개발자 안내서의 Global Accelerator에서 고유 IP 주소 가져오기를 참조하세요. AWS

- 자세한 API 내용은 명령 참조ProvisionByoipCidr의 섹션을 참조하세요. AWS CLI

### `tag-resource`

다음 코드 예시에서는 `tag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

액셀러레이터에 태그를 지정하려면

다음 `tag-resource` 예제에서는 각각에 해당하는 값과 함께 태그 이름 및 프로젝트를 액셀러레이터에 추가합니다.

```
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"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Global Accelerator 개발자 안내서의 Global Accelerator에서 태그 지정을 참조하세요. AWS

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### `untag-resource`

다음 코드 예시에서는 `untag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

액셀러레이터에서 태그를 제거하려면

다음 untag-resource 예제에서는 가속기에서 태그 이름 및 프로젝트를 제거합니다.

```
aws globalaccelerator untag-resource \
    --resource-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh \
    --tag-keys Key="Name" Key="Project"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Global Accelerator 개발자 안내서의 Global Accelerator에서 태그 지정을 참조하세요. AWS

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-accelerator-attributes

다음 코드 예시에서는 update-accelerator-attributes을 사용하는 방법을 보여 줍니다.

AWS CLI

액셀러레이터의 속성을 업데이트하려면

다음 update-accelerator-attributes 예제에서는 흐름 로그를 활성화하도록 액셀러레이터를 업데이트합니다. 가속기 속성을 생성하거나 업데이트하려면 US-West-2 (Oregon) 리전을 지정해야 합니다.

```
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
```

출력:

```
{
    "AcceleratorAttributes": {
        "FlowLogsEnabled": true
        "FlowLogsS3Bucket": flowlogs-abc
        "FlowLogsS3Prefix": bucketprefix-abc
    }
```

```
}
```

자세한 내용은 AWS Global Accelerator 개발자 안내서의 Global Accelerator의 AWS Accelerator를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateAcceleratorAttributes의 섹션을 참조하세요. AWS CLI

### update-accelerator

다음 코드 예시에서는 update-accelerator을 사용하는 방법을 보여 줍니다.

AWS CLI

액셀러레이터를 업데이트하려면

다음 update-accelerator 예제에서는 액셀러레이터 이름을 로 변경하도록 액셀러레이터를 수정합니다ExampleAcceleratorNew. 액셀러레이터를 생성하거나 업데이트하려면 US-West-2 (Oregon) 리전을 지정해야 합니다.

```
aws globalaccelerator update-accelerator \
    --accelerator-arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh \
    --name ExampleAcceleratorNew
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 AWS Global Accelerator 개발자 안내서의 Global Accelerator의 AWS Accelerator를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateAccelerator의 섹션을 참조하세요. AWS CLI

### update-custom-routing-accelerator-attributes

다음 코드 예시에서는 update-custom-routing-accelerator-attributes을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 라우팅 액셀러레이터의 속성을 업데이트하려면

다음 update-custom-routing-accelerator-attributes 예제에서는 흐름 로그를 활성화하도록 사용자 지정 라우팅 액셀러레이터를 업데이트합니다.

```
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
```

출력:

```
{
    "AcceleratorAttributes": {
        "FlowLogsEnabled": true
        "FlowLogsS3Bucket": flowlogs-abc
        "FlowLogsS3Prefix": bucketprefix-abc
    }
}
```

자세한 내용은 AWS Global Accelerator 개발자 안내서의 Global Accelerator에서 사용자 지정 라우팅 액셀러레이터를 참조하세요. AWS

- 자세한 API 내용은 명령 참조UpdateCustomRoutingAcceleratorAttributes의 섹션을 참조하세요. AWS CLI

### update-custom-routing-accelerator

다음 코드 예시에서는 update-custom-routing-accelerator을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 라우팅 액셀러레이터를 업데이트하려면

다음 update-custom-routing-accelerator 예제에서는 사용자 지정 라우팅 액셀러레이터를 수정하여 액셀러레이터 이름을 변경합니다.

```
aws globalaccelerator --region us-west-2 update-custom-routing-accelerator \
    --accelerator-arn arn:aws:globalaccelerator::123456789012:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh \
    --name ExampleCustomRoutingAcceleratorNew
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 AWS Global Accelerator 개발자 안내서의 Global Accelerator에서 사용자 지정 라우팅 액셀러레이터를 참조하세요. AWS

- 자세한 API 내용은 명령 참조UpdateCustomRoutingAccelerator의 섹션을 참조하세요. AWS CLI

### update-custom-routing-listener

다음 코드 예시에서는 update-custom-routing-listener을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 라우팅 액셀러레이터의 리스너를 업데이트하려면

다음 update-custom-routing-listener 예제에서는 리스너를 업데이트하여 포트 범위를 변경합니다.

```
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
```

출력:

```
{
    "Listener": {
        "ListenerArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-abcd-1234abcdefgh/listener/0123vxyz
        "PortRanges": [
            {
                "FromPort": 10000,
                "ToPort": 20000
            }
        ],
        "Protocol": "TCP"
    }
}
```

자세한 내용은 AWS Global Accelerator 개발자 안내서의 Global Accelerator에서 사용자 지정 라우팅 액셀러레이터용 리스너를 참조하세요. AWS

- 자세한 API 내용은 명령 참조UpdateCustomRoutingListener의 섹션을 참조하세요. AWS CLI

### update-endpoint-group

다음 코드 예시에서는 update-endpoint-group을 사용하는 방법을 보여 줍니다.

AWS CLI

엔드포인트 그룹을 업데이트하려면

다음 update-endpoint-group 예제에서는 엔드포인트 그룹에 탄력적 IP 주소, 및 ALB의 세 가지 엔드포인트를 추가합니다NLB.

```
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/
ab88888example \
    --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
```

출력:

```
{
    "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-4321abcdefg",
        "EndpointGroupRegion": "us-east-1"
    }
}
```

자세한 내용은 AWS Global Accelerator 개발자 안내서의 Global Accelerator의 엔드포인트 그룹을 참조하세요. AWS

- 자세한 API 내용은 명령 참조UpdateEndpointGroup의 섹션을 참조하세요. AWS CLI

### update-listener

다음 코드 예시에서는 update-listener을 사용하는 방법을 보여 줍니다.

AWS CLI

리스너를 업데이트하려면

다음 update-listener 예제에서는 리스너를 업데이트하여 포트를 100으로 변경합니다.

```
aws globalaccelerator update-listener \
    --listener-arn arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-
abcd-1234-abcd-1234abcdefgh/listener/0123vxyz \
    --port-ranges FromPort=100,ToPort=100
```

출력:

```
{
    "Listener": {
        "ListenerArn":
 "arn:aws:globalaccelerator::012345678901:accelerator/1234abcd-abcd-1234-
abcd-1234abcdefgh/listener/0123vxyz
        "PortRanges": [
```

```
            {
                "FromPort": 100,
                "ToPort": 100
            }
        ],
        "Protocol": "TCP",
        "ClientAffinity": "NONE"
    }
}
```

자세한 내용은 AWS Global Accelerator 개발자 안내서의 Global Accelerator의 리스너를 참조하세요. AWS

- 자세한 API 내용은 명령 참조UpdateListener의 섹션을 참조하세요. AWS CLI

### `withdraw-byoip-cidr`

다음 코드 예시에서는 `withdraw-byoip-cidr`을 사용하는 방법을 보여 줍니다.

AWS CLI

주소 범위를 철회하려면

다음 `withdraw-byoip-cidr` 예제에서는 이전에 AWS 리소스와 함께 사용하도록 광고한 AWS Global Accelerator에서 주소 범위를 철회합니다.

```
aws globalaccelerator withdraw-byoip-cidr \
    --cidr 192.0.2.250/24
```

출력:

```
{
    "ByoipCidr": {
        "Cidr": "192.0.2.250/24",
        "State": "PENDING_WITHDRAWING"
    }
}
```

자세한 내용은 AWS Global Accelerator 개발자 안내서의 Global Accelerator에서 고유 IP 주소 가져오기를 참조하세요. AWS

- 자세한 API 내용은 명령 참조WithdrawByoipCidr의 섹션을 참조하세요. AWS CLI

# AWS Glue 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS Glue.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### batch-stop-job-run

다음 코드 예시에서는 batch-stop-job-run을 사용하는 방법을 보여 줍니다.

AWS CLI

작업 실행을 중지하려면

다음 batch-stop-job-run 예제에서는 작업 실행을 중지합니다.

```
aws glue batch-stop-job-run \
    --job-name "my-testing-job" \
    --job-run-id jr_852f1de1f29fb62e0ba4166c33970803935d87f14f96cfdee5089d5274a61d3f
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 AWS Glue 개발자 안내서의 작업 실행을 참조하세요.

- 자세한 API 내용은 명령 참조BatchStopJobRun의 섹션을 참조하세요. AWS CLI

### `create-connection`

다음 코드 예시에서는 `create-connection`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS Glue 데이터 스토어에 대한 연결을 생성하려면

다음 `create-connection` 예제에서는 Kafka 데이터 스토어에 대한 연결 정보를 제공하는 연결을 AWS Glue 데이터 카탈로그에 생성합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Glue 개발자 안내서의 Glue 데이터 카탈로그에서 연결 정의를 참조하세요. AWS

- 자세한 API 내용은 명령 참조CreateConnection의 섹션을 참조하세요. AWS CLI

### create-database

다음 코드 예시에서는 create-database을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터베이스를 생성하려면

다음 create-database 예제에서는 AWS Glue 데이터 카탈로그에 데이터베이스를 생성합니다.

```
aws glue create-database \
    --database-input "{\"Name\":\"tempdb\"}" \
    --profile my_profile \
    --endpoint https://glue.us-east-1.amazonaws.com
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Glue 개발자 가이드의 데이터 카탈로그에서 데이터베이스 정의를 참조하세요.

- 자세한 API 내용은 명령 참조CreateDatabase의 섹션을 참조하세요. AWS CLI

### create-job

다음 코드 예시에서는 create-job을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터를 변환하는 작업을 생성하려면

다음 create-job 예제는 S3에 저장된 스크립트를 실행하는 스트리밍 작업을 생성합니다.

```
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
```

test_script.scala의 콘텐츠:

```
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()
    }
}
```

출력:

```
{
    "Name": "my-testing-job"
}
```

자세한 내용은 AWS Glue 개발자 안내서의 Glue에서 작업 작성을 참조하세요. AWS

- 자세한 API 내용은 명령 참조CreateJob의 섹션을 참조하세요. AWS CLI

### create-table

다음 코드 예시에서는 create-table을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: Kinesis 데이터 스트림에 대한 테이블 생성

다음 create-table 예제에서는 AWS Glue Data Catalog에 Kinesis 데이터 스트림을 설명하는 테이블을 생성합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Glue 개발자 안내서의 Glue 데이터 카탈로그의 테이블 정의를 참조하세요. AWS

예제 2: Kafka 데이터 스토어에 대한 테이블 생성

다음 `create-table` 예제에서는 Kafka 데이터 스토어를 설명하는 테이블을 AWS Glue Data Catalog에 생성합니다.

```
aws glue create-table \
    --database-name tempdb \
    --table-input  '{"Name":"test-kafka-input", "StorageDescriptor":{ \
            "Columns":[ \
                {"Name":"sensorid", "Type":"int"}, \
                {"Name":"currenttemperature", "Type":"int"}, \
                {"Name":"status", "Type":"string"} \
            ], \
            "Location":"glue-topic", \
            "Parameters":{ \
                "typeOfData":"kafka","topicName":"glue-topic", \
                "connectionName":"my-kafka-connection" \
            }, \
            "SerdeInfo":{ \
                "SerializationLibrary":"org.apache.hadoop.hive.serde2.OpenCSVSerde"}
 \
        }, \
        "Parameters":{ \
            "separatorChar":","} \
        }' \
    --profile my-profile \
    --endpoint https://glue.us-east-1.amazonaws.com
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Glue 개발자 안내서의 Glue 데이터 카탈로그의 테이블 정의를 참조하세요. AWS

예제 3: AWS S3 데이터 스토어에 대한 테이블 생성

다음 `create-table` 예제에서는 AWS Glue 데이터 카탈로그에 AWS Simple Storage Service(AWS S3) 데이터 스토어를 설명하는 테이블을 생성합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Glue 개발자 안내서의 Glue 데이터 카탈로그의 테이블 정의를 참조하세요. AWS

- 자세한 API 내용은 명령 참조CreateTable의 섹션을 참조하세요. AWS CLI

### delete-job

다음 코드 예시에서는 delete-job을 사용하는 방법을 보여 줍니다.

AWS CLI

작업을 삭제하려면

다음 delete-job 예제에서는 더 이상 필요하지 않은 작업을 삭제합니다.

```
aws glue delete-job \
    --job-name my-testing-job
```

출력:

```
{
    "JobName": "my-testing-job"
}
```

자세한 내용은 AWS Glue 개발자 안내서의 Glue 콘솔에서 작업 작업을 참조하세요. AWS

- 자세한 API 내용은 명령 참조DeleteJob의 섹션을 참조하세요. AWS CLI

### get-databases

다음 코드 예시에서는 get-databases을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS Glue 데이터 카탈로그에 일부 또는 모든 데이터베이스의 정의를 나열하려면

다음 get-databases 예제는 데이터 카탈로그의 데이터베이스에 대한 정보를 반환합니다.

```
aws glue get-databases
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS Glue 개발자 가이드의 데이터 카탈로그에서 데이터베이스 정의를 참조하세요.

- 자세한 API 내용은 명령 참조GetDatabases의 섹션을 참조하세요. AWS CLI

### get-job-run

다음 코드 예시에서는 get-job-run을 사용하는 방법을 보여 줍니다.

AWS CLI

작업 실행 정보를 가져오려면

다음 get-job-run 예제는 작업 실행 정보를 검색합니다.

```
aws glue get-job-run \
    --job-name "Combine legistators data" \
    --run-id jr_012e176506505074d94d761755e5c62538ee1aad6f17d39f527e9140cf0c9a5e
```

출력:

```
{
    "JobRun": {
        "Id": "jr_012e176506505074d94d761755e5c62538ee1aad6f17d39f527e9140cf0c9a5e",
        "Attempt": 0,
        "JobName": "Combine legistators 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"
    }
```

```
}
```

자세한 내용은 AWS Glue 개발자 안내서의 작업 실행을 참조하세요.

- 자세한 API 내용은 명령 참조GetJobRun의 섹션을 참조하세요. AWS CLI

### get-job-runs

다음 코드 예시에서는 get-job-runs을 사용하는 방법을 보여 줍니다.

AWS CLI

작업에 대한 모든 작업 실행 정보를 가져오려면

다음 get-job-runs 예제는 작업에 대한 작업 실행 정보를 검색합니다.

```
aws glue get-job-runs \
    --job-name "my-testing-job"
```

출력:

```
{
    "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/
TXtuGOqfByajkfL/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
+PpC6GpORahyF2fp9flB5SSb2bTGPnUSPVizLXRl1PN3QZldb+v1o9qRVktNYbW8=)",
            "PredecessorRuns": [],
            "AllocatedCapacity": 10,
            "ExecutionTime": 113,
            "Timeout": 2880,
            "MaxCapacity": 10.0,
            "WorkerType": "G.1X",
            "NumberOfWorkers": 10,
            "LogGroupName": "/aws-glue/jobs",
            "GlueVersion": "2.0"
        }
    ]
}
```

자세한 내용은 AWS Glue 개발자 안내서의 작업 실행을 참조하세요.

- 자세한 API 내용은 명령 참조GetJobRuns의 섹션을 참조하세요. AWS CLI

### get-job

다음 코드 예시에서는 get-job을 사용하는 방법을 보여 줍니다.

AWS CLI

작업 정보를 검색하려면

다음 get-job 예제는 작업 정보를 검색합니다.

```
aws glue get-job \
    --job-name my-testing-job
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS Glue 개발자 안내서의 작업을 참조하세요.

- 자세한 API 내용은 명령 참조GetJob의 섹션을 참조하세요. AWS CLI

### get-plan

다음 코드 예시에서는 get-plan을 사용하는 방법을 보여 줍니다.

AWS CLI

소스 테이블에서 대상 테이블로 데이터를 매핑하기 위해 생성된 코드를 가져오려면

다음은 데이터 소스에서 데이터 대상으로 열을 매핑하기 위해 생성된 코드를 get-plan 검색합니다.

```
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"
```

출력:

```
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()
  }
}
```

자세한 내용은 AWS Glue 개발자 안내서의 Glue에서 스크립트 편집을 참조하세요. AWS

- 자세한 API 내용은 명령 참조GetPlan의 섹션을 참조하세요. AWS CLI

### get-tables

다음 코드 예시에서는 get-tables을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 데이터베이스의 일부 또는 모든 테이블의 정의를 나열하려면

다음 get-tables 예제는 지정된 데이터베이스의 테이블에 대한 정보를 반환합니다.

```
aws glue get-tables --database-name 'tempdb'
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS Glue 개발자 안내서의 Glue 데이터 카탈로그의 테이블 정의를 참조하세요. AWS

- 자세한 API 내용은 명령 참조GetTables의 섹션을 참조하세요. AWS CLI

### `start-crawler`

다음 코드 예시에서는 `start-crawler`을 사용하는 방법을 보여 줍니다.

AWS CLI

크롤러를 시작하려면

다음 `start-crawler` 예제에서는 크롤러를 시작합니다.

```
aws glue start-crawler --name my-crawler
```

출력:

```
None
```

자세한 내용은 AWS Glue 개발자 안내서의 크롤러 정의 섹션을 참조하세요.

- 자세한 API 내용은 명령 참조StartCrawler의 섹션을 참조하세요. AWS CLI

**start-job-run**

다음 코드 예시에서는 start-job-run을 사용하는 방법을 보여 줍니다.

AWS CLI

작업을 실행하기 시작하려면

다음 start-job-run 예제에서는 작업을 시작합니다.

```
aws glue start-job-run \
    --job-name my-job
```

출력:

```
{
    "JobRunId":
 "jr_22208b1f44eb5376a60569d4b21dd20fcb8621e1a366b4e7b2494af764b82ded"
}
```

자세한 내용은 AWS Glue 개발자 안내서의 작업 작성을 참조하세요.

- 자세한 API 내용은 명령 참조StartJobRun의 섹션을 참조하세요. AWS CLI

## GuardDuty 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 GuardDuty.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### `accept-invitation`

다음 코드 예시에서는 `accept-invitation`을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 리전의 GuardDuty 멤버 계정이 되기 위한 초대를 수락하려면

다음 `accept-invitation` 예제에서는 현재 리전에서 GuardDuty 멤버 계정이 되기 위한 초대를 수락하는 방법을 보여줍니다.

```
aws guardduty accept-invitation  \
    --detector-id 12abc34d567e8fa901bc2d34eexample \
    --master-id 123456789111 \
    --invitation-id d6b94fb03a66ff665f7db8764example
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 GuardDuty 사용 설명서의 초대를 통한 GuardDuty 계정 관리를 참조하세요.

- 자세한 API 내용은 명령 참조AcceptInvitation의 섹션을 참조하세요. AWS CLI

### `archive-findings`

다음 코드 예시에서는 `archive-findings`을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 리전에서 조사 결과를 아카이브하려면

이 예제는 현재 리전에서 조사 결과를 아카이브하는 방법을 보여줍니다.

```
aws guardduty archive-findings \
    --detector-id 12abc34d567e8fa901bc2d34eexample \
    --finding-ids d6b94fb03a66ff665f7db8764example 3eb970e0de00c16ec14e6910fexample
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 GuardDuty 사용 설명서의 초대를 통한 GuardDuty 계정 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ArchiveFindings의 섹션을 참조하세요. AWS CLI

### create-detector

다음 코드 예시에서는 create-detector을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 리전 GuardDuty 에서 를 활성화하려면

이 예제는 현재 리전 GuardDuty에서 를 활성화하는 새 탐지기를 생성하는 방법을 보여줍니다.

```
aws guardduty create-detector \
    --enable
```

출력:

```
{
    "DetectorId": "b6b992d6d2f48e64bc59180bfexample"
}
```

자세한 내용은 GuardDuty 사용 설명서의 Amazon 활성화 GuardDuty를 참조하세요.

- 자세한 API 내용은 명령 참조CreateDetector의 섹션을 참조하세요. AWS CLI

### create-filter

다음 코드 예시에서는 create-filter을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 리전에 대한 새 필터를 생성하려면

이 예제에서는 특정 이미지에서 생성된 인스턴스의 모든 포트 스캔 결과와 일치하는 필터를 생성합니다.

```
aws guardduty create-filter \
    --detector-id b6b992d6d2f48e64bc59180bfexample \
    --action ARCHIVE \
    --name myFilter \
    --finding-criteria '{"Criterion": {"type": {"Eq": ["Recon:EC2/
Portscan"]},"resource.instanceDetails.imageId": {"Eq": ["ami-0a7a207083example"]}}}'
```

출력:

```
{
    "Name": "myFilter"
}
```

자세한 내용은 GuardDuty 사용 설명서의 결과 필터링을 참조하세요.

- 자세한 API 내용은 명령 참조CreateFilter의 섹션을 참조하세요. AWS CLI

### create-ip-set

다음 코드 예시에서는 create-ip-set을 사용하는 방법을 보여 줍니다.

AWS CLI

신뢰할 수 있는 IP 세트를 생성하려면

다음 create-ip-set 예제에서는 현재 리전에서 신뢰할 수 있는 IP 세트를 생성하고 활성화합니다.

```
aws guardduty create-ip-set \
    --detector-id 12abc34d567e8fa901bc2d34eexample \
    --name new-ip-set \
    --format TXT
    --location s3://AWSDOC-EXAMPLE-BUCKET/customtrustlist.csv
    --activate
```

출력:

```
{
    "IpSetId": "d4b94fc952d6912b8f3060768example"
}
```

자세한 내용은 GuardDuty 사용 설명서의 신뢰할 수 있는 IP 목록 및 위협 목록 작업을 참조하세요.

- 자세한 API 내용은 명령 참조CreateIpSet의 섹션을 참조하세요. AWS CLI

### create-members

다음 코드 예시에서는 create-members을 사용하는 방법을 보여 줍니다.

AWS CLI

새 멤버를 현재 리전의 GuardDuty 마스터 계정과 연결합니다.

이 예제에서는 현재 계정에서 관리할 멤버 계정을 GuardDuty 마스터로 연결하는 방법을 보여줍니다.

```
aws guardduty create-members
    --detector-id b6b992d6d2f48e64bc59180bfexample \
    --account-details AccountId=111122223333,Email=first
+member@example.com AccountId=111111111111 ,Email=another+member@example.com
```

출력:

```
{
   "UnprocessedAccounts": []
}
```

자세한 내용은 GuardDuty 사용 설명서의 여러 계정 관리를 참조하세요.

- 자세한 API 내용은 명령 참조CreateMembers의 섹션을 참조하세요. AWS CLI

### `create-publishing-destination`

다음 코드 예시에서는 `create-publishing-destination`을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 리전의 GuardDuty 결과를 내보낼 게시 대상을 생성합니다.

이 예제에서는 GuardDuty 조사 결과에 대한 게시 대상을 생성하는 방법을 보여줍니다.

```
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
```

출력:

```
{
```

```
    "DestinationId": "46b99823849e1bbc242dfbe3cexample"
}
```

자세한 내용은 GuardDuty 사용 설명서의 결과 내보내기를 참조하세요.

- 자세한 API 내용은 명령 참조CreatePublishingDestination의 섹션을 참조하세요. AWS CLI

### create-sample-findings

다음 코드 예시에서는 create-sample-findings을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 리전에서 샘플 GuardDuty 조사 결과를 생성합니다.

이 예제에서는 제공된 유형의 샘플 결과를 생성하는 방법을 보여줍니다.

```
aws guardduty create-sample-findings \
    --detector-id b6b992d6d2f48e64bc59180bfexample \
    --finding-types UnauthorizedAccess:EC2/TorClient UnauthorizedAccess:EC2/TorRelay
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 GuardDuty 사용 설명서의 샘플 조사 결과를 참조하세요.

- 자세한 API 내용은 명령 참조CreateSampleFindings의 섹션을 참조하세요. AWS CLI

### create-threat-intel-set

다음 코드 예시에서는 create-threat-intel-set을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 리전에서 새 위협 인텔 세트를 생성합니다.

이 예제에서는 위협 인텔 세트를 에 업로드 GuardDuty 하고 즉시 활성화하는 방법을 보여줍니다.

```
aws guardduty create-threat-intel-set \
    --detector-id b6b992d6d2f48e64bc59180bfexample \
    --name myThreatSet \
    --format TXT \
    --location s3://EXAMPLEBUCKET/threatlist.csv \
```

```
    --activate
```

출력:

```
{
    "ThreatIntelSetId": "20b9a4691aeb33506b808878cexample"
}
```

자세한 내용은 GuardDuty 사용 설명서의 신뢰할 수 있는 IP 및 위협 목록을 참조하세요.

- 자세한 API 내용은 명령 참조CreateThreatIntelSet의 섹션을 참조하세요. AWS CLI

### `decline-invitations`

다음 코드 예시에서는 `decline-invitations`을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 리전의 다른 계정에서 Guardduty를 관리하도록 초대를 거부합니다.

이 예제에서는 멤버십 초대를 거부하는 방법을 보여줍니다.

```
aws guardduty decline-invitations \
    --account-ids 111122223333
```

출력:

```
{
    "UnprocessedAccounts": []
}
```

자세한 내용은 GuardDuty 사용 설명서의 초대를 통한 GuardDuty 계정 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DeclineInvitations의 섹션을 참조하세요. AWS CLI

### `delete-detector`

다음 코드 예시에서는 `delete-detector`을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 리전 GuardDuty에서 탐지기를 삭제하고 를 비활성화합니다.

이 예제에서는 탐지기를 삭제하는 방법을 보여줍니다. 성공하면 해당 탐지기와 연결된 리전 GuardDuty 에서 비활성화됩니다.

```
aws guardduty delete-detector \
    --detector-id b6b992d6d2f48e64bc59180bfexample
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 GuardDuty 사용 설명서의 일시 중지 또는 비활성화 GuardDuty를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteDetector의 섹션을 참조하세요. AWS CLI

### `delete-filter`

다음 코드 예시에서는 `delete-filter`을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 리전에서 기존 필터를 삭제하려면

이 예제에서는 필터 삭제를 생성하는 방법을 보여줍니다.

```
aws guardduty delete-filter \
    --detector-id b6b992d6d2f48e64bc59180bfexample \
    --filter-name byebyeFilter
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 GuardDuty 사용 설명서의 결과 필터링을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteFilter의 섹션을 참조하세요. AWS CLI

### `disable-organization-admin-account`

다음 코드 예시에서는 `disable-organization-admin-account`을 사용하는 방법을 보여 줍니다.

AWS CLI

조직 GuardDuty 내에서 의 위임된 관리자로 계정을 제거하려면

이 예제에서는 의 위임된 관리자로서 계정을 제거하는 방법을 보여줍니다 GuardDuty.

```
aws guardduty disable-organization-admin-account \
```

```
    --admin-account-id 111122223333
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 GuardDuty 사용 설명서의 AWS 조직 계정 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DisableOrganizationAdminAccount의 섹션을 참조하세요. AWS CLI

### disassociate-from-master-account

다음 코드 예시에서는 disassociate-from-master-account을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 리전의 현재 마스터 계정에서 연결을 해제하려면

다음 disassociate-from-master-account 예제에서는 현재 AWS 리전의 현재 GuardDuty 마스터 계정에서 계정 연결을 해제합니다.

```
aws guardduty disassociate-from-master-account \
    --detector-id d4b040365221be2b54a6264dcexample
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 GuardDuty 사용 설명서의 GuardDuty 마스터 계정과 멤버 계정 간의 관계 이해를 참조하세요.

- 자세한 API 내용은 명령 참조DisassociateFromMasterAccount의 섹션을 참조하세요. AWS CLI

### get-detector

다음 코드 예시에서는 get-detector을 사용하는 방법을 보여 줍니다.

AWS CLI

특정 감지기의 세부 정보를 검색하려면

다음 get-detector 예제에서는 지정된 감지기의 구성 세부 정보를 표시합니다.

```
aws guardduty get-detector \
    --detector-id 12abc34d567e8fa901bc2d34eexample
```

출력:

```
{
    "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"
}
```

자세한 내용은 GuardDuty 사용 설명서의 개념 및 용어를 참조하세요.

- 자세한 API 내용은 명령 참조GetDetector의 섹션을 참조하세요. AWS CLI

### get-findings

다음 코드 예시에서는 get-findings을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 특정 결과의 세부 정보를 검색하려면

다음 get-findings 예제에서는 지정된 JSON 결과의 전체 결과 세부 정보를 검색합니다.

```
aws guardduty get-findings \
    --detector-id 12abc34d567e8fa901bc2d34eexample \
    --finding-id 1ab92989eaf0e742df4a014d5example
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 GuardDuty 사용 설명서의 결과를 참조하세요.

- 자세한 API 내용은 명령 참조GetFindings의 섹션을 참조하세요. AWS CLI

### get-ip-set

다음 코드 예시에서는 get-ip-set을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 신뢰할 수 있는 IP 세트에 대한 세부 정보를 나열하려면

다음 get-ip-set 예제에서는 지정된 신뢰할 수 있는 IP 세트의 상태와 세부 정보를 보여줍니다.

```
aws guardduty get-ip-set \
    --detector-id 12abc34d567e8fa901bc2d34eexample \
    --ip-set-id d4b94fc952d6912b8f3060768example
```

출력:

```
{
    "Status": "ACTIVE",
    "Location": "s3://AWSDOC-EXAMPLE-BUCKET.s3-us-west-2.amazonaws.com/
customlist.csv",
    "Tags": {},
    "Format": "TXT",
    "Name": "test-ip-set"
}
```

자세한 내용은 GuardDuty 사용 설명서의 신뢰할 수 있는 IP 목록 및 위협 목록 작업을 참조하세요.

- 자세한 API 내용은 명령 참조GetIpSet의 섹션을 참조하세요. AWS CLI

### get-master-account

다음 코드 예시에서는 get-master-account을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 리전에서 마스터 계정에 대한 세부 정보를 검색하려면

다음 get-master-account 예제에서는 현재 리전의 감지기와 연결된 마스터 계정의 상태와 세부 정보를 표시합니다.

```
aws guardduty get-master-account \
    --detector-id 12abc34d567e8fa901bc2d34eexample
```

출력:

```
{
    "Master": {
        "InvitationId": "04b94d9704854a73f94e061e8example",
        "InvitedAt": "2020-06-09T22:23:04.970Z",
        "RelationshipStatus": "Enabled",
        "AccountId": "123456789111"
    }
}
```

자세한 내용은 GuardDuty 사용 설명서의 GuardDuty 마스터 계정과 멤버 계정 간의 관계 이해를 참조하세요.

- 자세한 API 내용은 명령 참조GetMasterAccount의 섹션을 참조하세요. AWS CLI

### list-detectors

다음 코드 예시에서는 list-detectors을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 리전에서 사용 가능한 감지기를 나열하려면

다음 list-detectors 예제에서는 현재 AWS 리전에서 사용 가능한 감지기를 나열합니다.

```
aws guardduty list-detectors
```

출력:

```
{
    "DetectorIds": [
        "12abc34d567e8fa901bc2d34eexample"
    ]
}
```

자세한 내용은 GuardDuty 사용 설명서의 개념 및 용어를 참조하세요.

- 자세한 API 내용은 명령 참조ListDetectors의 섹션을 참조하세요. AWS CLI

### `list-findings`

다음 코드 예시에서는 `list-findings`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 현재 리전에 대한 모든 조사 결과를 나열하려면

다음 `list-findings` 예제에서는 심각도를 기준으로 가장 높음에서 가장 낮음으로 정렬된 findingIds 현재 리전의 전체 목록을 표시합니다.

```
aws guardduty list-findings \
    --detector-id 12abc34d567e8fa901bc2d34eexample \
    --sort-criteria '{"AttributeName": "severity","OrderBy":"DESC"}'
```

출력:

```
{
    "FindingIds": [
        "04b8ab50fd29c64fc771b232dexample",
        "5ab8ab50fd21373735c826d3aexample",
        "90b93de7aba69107f05bbe60bexample",
        ...
    ]
}
```

자세한 내용은 GuardDuty 사용 설명서의 결과를 참조하세요.

예제 2: 특정 결과 기준과 일치하는 현재 리전의 결과를 나열하려면

다음 `list-findings` 예제에서는 지정된 결과 유형 findingIds 과 일치하는 모든 목록의 목록을 표시합니다.

```
aws guardduty list-findings \
    --detector-id 12abc34d567e8fa901bc2d34eexample \
    --finding-criteria  '{"Criterion":{"type": {"Eq":["UnauthorizedAccess:EC2/
SSHBruteForce"]}}}'
```

출력:

```
{
    "FindingIds": [
        "90b93de7aba69107f05bbe60bexample",
        "6eb9430d7023d30774d6f05e3example",
        "2eb91a2d060ac9a21963a5848example",
        "44b8ab50fd2b0039a9e48f570example",
        "9eb8ab4cd2b7e5b66ba4f5e96example",
        "e0b8ab3a38e9b0312cc390ceeexample"
    ]
}
```

자세한 내용은 GuardDuty 사용 설명서의 결과를 참조하세요.

예제 3: JSON 파일 내에 정의된 특정 결과 기준 집합과 일치하는 현재 리전의 결과를 나열하려면

다음 `list-findings` 예제에서는 JSON 파일에 지정된 대로 아카이브 findingIds 되지 않은 모든 목록과 'testuser'라는 IAM 사용자와 관련된 목록을 표시합니다.

```
aws guardduty list-findings \
    --detector-id 12abc34d567e8fa901bc2d34eexample \
    --finding-criteria  file://myfile.json
```

`myfile.json`의 콘텐츠:

```
{"Criterion": {
    "resource.accessKeyDetails.userName":{
```

```
                "Eq":[
                    "testuser"
                    ]
                },
    "service.archived": {
                "Eq": [
                    "false"
                ]
            }
        }
}
```

출력:

```
{
    "FindingIds": [
        "1ab92989eaf0e742df4a014d5example"
    ]
}
```

자세한 내용은 GuardDuty 사용 설명서의 결과를 참조하세요.

- 자세한 API 내용은 명령 참조ListFindings의 섹션을 참조하세요. AWS CLI

### `list-invitations`

다음 코드 예시에서는 `list-invitations`을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 리전의 멤버 계정이 되기 위한 초대에 대한 세부 정보를 나열하려면

다음 `list-invitations` 예제에서는 현재 리전의 GuardDuty 멤버 계정이 되기 위한 초대의 세부 정보와 상태를 나열합니다.

```
aws guardduty list-invitations
```

출력:

```
{
    "Invitations": [
```

```
        {
            "InvitationId": "d6b94fb03a66ff665f7db8764example",
            "InvitedAt": "2020-06-10T17:56:38.221Z",
            "RelationshipStatus": "Invited",
            "AccountId": "123456789111"
        }
    ]
}
```

자세한 내용은 GuardDuty 사용 설명서의 초대를 통한 GuardDuty 계정 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ListInvitations의 섹션을 참조하세요. AWS CLI

### `list-ip-sets`

다음 코드 예시에서는 `list-ip-sets`을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 리전에서 신뢰할 수 있는 IP 세트를 나열하려면

다음 `list-ip-sets` 예제에서는 현재 AWS 리전의 신뢰할 수 있는 IP 세트를 나열합니다.

```
aws guardduty list-ip-sets \
    --detector-id 12abc34d567e8fa901bc2d34eexample
```

출력:

```
{
    "IpSetIds": [
        "d4b94fc952d6912b8f3060768example"
    ]
}
```

자세한 내용은 GuardDuty 사용 설명서의 신뢰할 수 있는 IP 목록 및 위협 목록 작업을 참조하세요.

- 자세한 API 내용은 명령 참조ListIpSets의 섹션을 참조하세요. AWS CLI

### `list-members`

다음 코드 예시에서는 `list-members`을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 리전의 모든 멤버를 나열하려면

다음 `list-members` 예제에서는 현재 리전에 대한 모든 멤버 계정과 세부 정보를 나열합니다.

```
aws guardduty list-members \
    --detector-id 12abc34d567e8fa901bc2d34eexample
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 GuardDuty 사용 설명서의 GuardDuty 마스터 계정과 멤버 계정 간의 관계 이해를 참조하세요.

- 자세한 API 내용은 명령 참조ListMembers의 섹션을 참조하세요. AWS CLI

### update-ip-set

다음 코드 예시에서는 `update-ip-set`을 사용하는 방법을 보여 줍니다.

AWS CLI

신뢰할 수 있는 IP 세트를 업데이트하려면

다음 `update-ip-set` 예제에서는 신뢰할 수 있는 IP 세트의 세부 정보를 업데이트하는 방법을 보여줍니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 GuardDuty 사용 설명서의 신뢰할 수 있는 IP 목록 및 위협 목록 작업을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateIpSet의 섹션을 참조하세요. AWS CLI

## AWS Health 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS Health.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `describe-affected-entities`

다음 코드 예시에서는 `describe-affected-entities`을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 AWS 상태 이벤트의 영향을 받는 엔터티를 나열하려면

다음 `describe-affected-entities` 예제에서는 지정된 AWS 상태 이벤트의 영향을 받는 엔터티를 나열합니다. 이 이벤트는 AWS 계정에 대한 결제 알림입니다.

```
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
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 AWS 상태 사용 설명서의 이벤트 로그를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeAffectedEntities의 섹션을 참조하세요. AWS CLI

### describe-event-details

다음 코드 예시에서는 describe-event-details을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 상태 이벤트에 대한 정보를 나열하려면

다음 describe-event-details 예제에서는 지정된 AWS 상태 이벤트에 대한 정보를 나열합니다.

```
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
```

출력:

```
{
```

```
    "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": []
}
```

자세한 내용은 AWS 상태 사용 설명서의 이벤트 세부 정보 창을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeEventDetails의 섹션을 참조하세요. AWS CLI

### `describe-events`

다음 코드 예시에서는 `describe-events`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: AWS 상태 이벤트 나열

다음 `describe-events` 예제에서는 최근 AWS 상태 이벤트를 나열합니다.

```
aws health describe-events \
    --region us-east-1
```

출력:


```
{
    "events": [
        {
            "arn": "arn:aws:health:us-west-1::event/ECS/AWS_ECS_OPERATIONAL_ISSUE/
AWS_ECS_OPERATIONAL_ISSUE_KWQPY_EXAMPLE111",
            "service": "ECS",
            "eventTypeCode": "AWS_ECS_OPERATIONAL_ISSUE",
            "eventTypeCategory": "issue",
            "region": "us-west-1",
            "startTime": 1589077890.53,
            "endTime": 1589086345.597,
            "lastUpdatedTime": 1589086345.905,
            "statusCode": "closed",
            "eventScopeCode": "PUBLIC"
        },
        {
            "arn": "arn:aws:health:global::event/BILLING/AWS_BILLING_NOTIFICATION/
AWS_BILLING_NOTIFICATION_6ce1d874-e995-40e2-99cd-EXAMPLE1118b",
            "service": "BILLING",
            "eventTypeCode": "AWS_BILLING_NOTIFICATION",
            "eventTypeCategory": "accountNotification",
            "region": "global",
            "startTime": 1588356000.0,
            "lastUpdatedTime": 1588356524.358,
            "statusCode": "open",
            "eventScopeCode": "ACCOUNT_SPECIFIC"
        },
        {
            "arn": "arn:aws:health:us-west-2::event/
CLOUDFORMATION/AWS_CLOUDFORMATION_OPERATIONAL_ISSUE/
AWS_CLOUDFORMATION_OPERATIONAL_ISSUE_OHTWY_EXAMPLE111",
            "service": "CLOUDFORMATION",
            "eventTypeCode": "AWS_CLOUDFORMATION_OPERATIONAL_ISSUE",
            "eventTypeCategory": "issue",
            "region": "us-west-2",
            "startTime": 1588279630.761,
            "endTime": 1588284650.0,
            "lastUpdatedTime": 1588284691.941,
            "statusCode": "closed",
            "eventScopeCode": "PUBLIC"
        },
        {
```

```
            "arn": "arn:aws:health:ap-northeast-1::event/LAMBDA/
AWS_LAMBDA_OPERATIONAL_ISSUE/AWS_LAMBDA_OPERATIONAL_ISSUE_JZDND_EXAMPLE111",
            "service": "LAMBDA",
            "eventTypeCode": "AWS_LAMBDA_OPERATIONAL_ISSUE",
            "eventTypeCategory": "issue",
            "region": "ap-northeast-1",
            "startTime": 1587379534.08,
            "endTime": 1587391771.0,
            "lastUpdatedTime": 1587395689.316,
            "statusCode": "closed",
            "eventScopeCode": "PUBLIC"
        },
        {
            "arn": "arn:aws:health:us-east-1::event/EC2/AWS_EC2_OPERATIONAL_ISSUE/
AWS_EC2_OPERATIONAL_ISSUE_COBXJ_EXAMPLE111",
            "service": "EC2",
            "eventTypeCode": "AWS_EC2_OPERATIONAL_ISSUE",
            "eventTypeCategory": "issue",
            "region": "us-east-1",
            "startTime": 1586473044.284,
            "endTime": 1586479706.091,
            "lastUpdatedTime": 1586479706.153,
            "statusCode": "closed",
            "eventScopeCode": "PUBLIC"
        },
        {
            "arn": "arn:aws:health:global::event/SECURITY/AWS_SECURITY_NOTIFICATION/
AWS_SECURITY_NOTIFICATION_42007387-8129-42da-8c88-EXAMPLE11139",
            "service": "SECURITY",
            "eventTypeCode": "AWS_SECURITY_NOTIFICATION",
            "eventTypeCategory": "accountNotification",
            "region": "global",
            "startTime": 1585674000.0,
            "lastUpdatedTime": 1585674004.132,
            "statusCode": "open",
            "eventScopeCode": "PUBLIC"
        },
        {
            "arn": "arn:aws:health:global::event/CLOUDFRONT/
AWS_CLOUDFRONT_OPERATIONAL_ISSUE/AWS_CLOUDFRONT_OPERATIONAL_ISSUE_FRQXG_EXAMPLE111",
            "service": "CLOUDFRONT",
            "eventTypeCode": "AWS_CLOUDFRONT_OPERATIONAL_ISSUE",
            "eventTypeCategory": "issue",
            "region": "global",
```

```
            "startTime": 1585610898.589,
            "endTime": 1585617671.0,
            "lastUpdatedTime": 1585620638.869,
            "statusCode": "closed",
            "eventScopeCode": "PUBLIC"
        },
        {
            "arn": "arn:aws:health:us-east-1::event/SES/AWS_SES_OPERATIONAL_ISSUE/
AWS_SES_OPERATIONAL_ISSUE_URNDF_EXAMPLE111",
            "service": "SES",
            "eventTypeCode": "AWS_SES_OPERATIONAL_ISSUE",
            "eventTypeCategory": "issue",
            "region": "us-east-1",
            "startTime": 1585342008.46,
            "endTime": 1585344017.0,
            "lastUpdatedTime": 1585344355.989,
            "statusCode": "closed",
            "eventScopeCode": "PUBLIC"
        },
        {
            "arn": "arn:aws:health:global::event/IAM/
AWS_IAM_OPERATIONAL_NOTIFICATION/
AWS_IAM_OPERATIONAL_NOTIFICATION_b6771c34-6ecd-4aea-9d3e-EXAMPLE1117e",
            "service": "IAM",
            "eventTypeCode": "AWS_IAM_OPERATIONAL_NOTIFICATION",
            "eventTypeCategory": "accountNotification",
            "region": "global",
            "startTime": 1584978300.0,
            "lastUpdatedTime": 1584978553.572,
            "statusCode": "open",
            "eventScopeCode": "ACCOUNT_SPECIFIC"
        },
        {
            "arn": "arn:aws:health:ap-southeast-2::event/EC2/
AWS_EC2_OPERATIONAL_ISSUE/AWS_EC2_OPERATIONAL_ISSUE_HNGHE_EXAMPLE111",
            "service": "EC2",
            "eventTypeCode": "AWS_EC2_OPERATIONAL_ISSUE",
            "eventTypeCategory": "issue",
            "region": "ap-southeast-2",
            "startTime": 1583881487.483,
            "endTime": 1583885056.785,
            "lastUpdatedTime": 1583885057.052,
            "statusCode": "closed",
            "eventScopeCode": "PUBLIC"
```

```
        }
    ]
}
```

자세한 내용은 상태 사용 설명서의 AWS Personal Health Dashboard 시작하기를 참조하세요. AWS

예제 2: 서비스 및 이벤트 상태 코드별로 AWS 상태 이벤트를 나열하려면

다음 `describe-events` 예제에서는 이벤트 상태가 종료된 Amazon Elastic Compute Cloud(Amazon EC2)의 AWS 상태 이벤트를 나열합니다.

```
aws health describe-events \
    --filter "services=EC2,eventStatusCodes=closed"
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 상태 사용 설명서의 AWS Personal Health Dashboard 시작하기를 참조하세요. AWS

- 자세한 API 내용은 명령 참조DescribeEvents의 섹션을 참조하세요. AWS CLI

## HealthImaging 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 HealthImaging.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### copy-image-set

다음 코드 예시에서는 copy-image-set을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 대상 없이 이미지 세트 복사

다음 copy-image-set 예제에서는 대상 없이 이미지 세트의 복사본을 만듭니다.

```
aws medical-imaging copy-image-set \
    --datastore-id 12345678901234567890123456789012 \
    --source-image-set-id ea92b0d8838c72a3f25d00d13616f87e \
    --copy-image-set-information '{"sourceImageSet": {"latestVersionId": "1" } }'
```

출력:

```
{
    "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"
}
```

예제 2: 대상과 함께 이미지 세트 복사

다음 copy-image-set 예제에서는 대상을 사용하여 이미지 세트의 복사본을 만듭니다.

```
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"} }'
```

출력:

```
{
    "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"
}
```

예제 3: 소스 이미지 세트의 인스턴스 하위 집합을 대상 이미지 세트로 복사합니다.

다음 copy-image-set 예제에서는 소스 이미지 세트의 DICOM 인스턴스 하나를 대상 이미지 세트로 복사합니다. 강제 파라미터는 환자, 연구 및 시리즈 수준 속성의 불일치를 재정의하기 위해 제공됩니다.

```
aws medical-imaging copy-image-set \
    --datastore-id 12345678901234567890123456789012 \
    --source-image-set-id ea92b0d8838c72a3f25d00d13616f87e \
    --copy-image-set-information '{"sourceImageSet": {"latestVersionId":
 "1","DICOMCopies": {"copiableAttributes": "{\"SchemaVersion\":\"1.1\",\"Study\":
{\"Series\":{\"1.3.6.1.4.1.5962.99.1.3673257865.2104868982.1369432891697.3666.0\":
{\"Instances\":
```

```
{\"1.3.6.1.4.1.5962.99.1.3673257865.2104868982.1369432891697.3669.0\":
{}}}}}}"}},"destinationImageSet": {"imageSetId":
 "b9eb50d8ee682eb9fcf4acbf92f62bb7","latestVersionId": "1"}}' \
    --force
```

출력:

```
{
    "destinationImageSetProperties": {
        "latestVersionId": "2",
        "imageSetWorkflowStatus": "COPYING",
        "updatedAt": 1680042505.135,
        "imageSetId": "b9eb50d8ee682eb9fcf4acbf92f62bb7",
        "imageSetState": "LOCKED",
        "createdAt": 1680042357.432
    },
    "sourceImageSetProperties": {
        "latestVersionId": "1",
        "imageSetWorkflowStatus": "COPYING_WITH_READ_ONLY_ACCESS",
        "updatedAt": 1680042505.135,
        "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
        "imageSetState": "LOCKED",
        "createdAt": 1680027126.436
    },
    "datastoreId": "12345678901234567890123456789012"
}
```

자세한 내용은 AWS HealthImaging 개발자 안내서의 이미지 세트 복사를 참조하세요.

- 자세한 API 내용은 명령 참조CopyImageSet의 섹션을 참조하세요. AWS CLI

### create-datastore

다음 코드 예시에서는 create-datastore을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 스토어 생성

다음은 이름이 my-datastore인 데이터 스토어를 생성하는 create-datastore 코드 예제입니다.

```
aws medical-imaging create-datastore \
```

```
    --datastore-name "my-datastore"
```

출력:

```
{
    "datastoreId": "12345678901234567890123456789012",
    "datastoreStatus": "CREATING"
}
```

자세한 내용은 AWS HealthImaging 개발자 안내서의 데이터 스토어 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateDatastore의 섹션을 참조하세요. AWS CLI

### delete-datastore

다음 코드 예시에서는 delete-datastore을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 스토어 삭제

다음은 데이터 스토어를 삭제하는 delete-datastore 코드 예제입니다.

```
aws medical-imaging delete-datastore \
    --datastore-id "12345678901234567890123456789012"
```

출력:

```
{
    "datastoreId": "12345678901234567890123456789012",
    "datastoreStatus": "DELETING"
}
```

자세한 내용은 AWS HealthImaging 개발자 안내서의 데이터 스토어 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteDatastore의 섹션을 참조하세요. AWS CLI

### delete-image-set

다음 코드 예시에서는 delete-image-set을 사용하는 방법을 보여 줍니다.

AWS CLI

이미지 세트 삭제

다음은 이미지 세트를 삭제하는 delete-image-set 코드 예제입니다.

```
aws medical-imaging delete-image-set \
    --datastore-id 12345678901234567890123456789012 \
    --image-set-id ea92b0d8838c72a3f25d00d13616f87e
```

출력:

```
{
    "imageSetWorkflowStatus": "DELETING",
    "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
    "imageSetState": "LOCKED",
    "datastoreId": "12345678901234567890123456789012"
}
```

자세한 내용은 AWS HealthImaging 개발자 안내서의 이미지 세트 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteImageSet의 섹션을 참조하세요. AWS CLI

### get-datastore

다음 코드 예시에서는 get-datastore을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 스토어 속성 가져오기

다음은 데이터 스토어 속성을 가져오는 get-datastore 코드 예제입니다.

```
aws medical-imaging get-datastore \
    --datastore-id 12345678901234567890123456789012
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS HealthImaging 개발자 안내서의 데이터 스토어 속성 가져오기를 참조하세요.

- 자세한 API 내용은 명령 참조GetDatastore의 섹션을 참조하세요. AWS CLI

### get-dicom-import-job

다음 코드 예시에서는 get-dicom-import-job을 사용하는 방법을 보여 줍니다.

AWS CLI

dicom 가져오기 작업의 속성 가져오기

다음은 dicom 가져오기 작업의 속성을 가져오는 get-dicom-import-job 코드 예제입니다.

```
aws medical-imaging get-dicom-import-job \
    --datastore-id "12345678901234567890123456789012" \
    --job-id "09876543210987654321098765432109"
```

출력:

```
{
    "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/"
    }
```

```
}
```

자세한 내용은 AWS HealthImaging 개발자 안내서의 가져오기 작업 속성 가져오기를 참조하세요.

- API 자세한 내용은 명령 참조의 G etDICOMImport작업을 참조하세요. AWS CLI

### get-image-frame

다음 코드 예시에서는 get-image-frame을 사용하는 방법을 보여 줍니다.

AWS CLI

이미지 세트 픽셀 데이터 가져오기

다음은 이미지 프레임을 가져오는 get-image-frame 코드 예제입니다.

```
aws medical-imaging get-image-frame \
    --datastore-id "12345678901234567890123456789012" \
    --image-set-id "98765412345612345678907890789012" \
    --image-frame-information imageFrameId=3abf5d5d7ae72f80a0ec81b2c0de3ef4 \
    imageframe.jph
```

참고: GetImageFrame 작업이 imageframe.jph 파일에 픽셀 데이터 스트림을 반환하기 때문에 이 코드 예제에는 출력이 포함되지 않습니다. 이미지 프레임 디코딩 및 보기에 대한 자세한 내용은 HTJ2K 디코딩 라이브러리를 참조하세요.

자세한 내용은 AWS HealthImaging 개발자 안내서의 이미지 세트 픽셀 데이터 가져오기를 참조하세요.

- 자세한 API 내용은 명령 참조GetImageFrame의 섹션을 참조하세요. AWS CLI

### get-image-set-metadata

다음 코드 예시에서는 get-image-set-metadata을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 버전 없이 이미지 세트 메타데이터 가져오기

다음은 버전을 지정하지 않고 이미지 세트의 메타데이터를 가져오는 get-image-set-metadata 코드 예제입니다.

참고: outfile은 필수 파라미터입니다.

```
aws medical-imaging get-image-set-metadata \
    --datastore-id 12345678901234567890123456789012 \
    --image-set-id ea92b0d8838c72a3f25d00d13616f87e \
    studymetadata.json.gz
```

반환된 메타데이터는 gzip으로 압축되어 studymetadata.json.gz 파일에 저장됩니다. 반환된 JSON 객체의 내용을 보려면 먼저 압축을 풀어야 합니다.

출력:

```
{
    "contentType": "application/json",
    "contentEncoding": "gzip"
}
```

예제 2: 버전과 함께 이미지 세트 메타데이터 가져오기

다음은 지정된 버전의 이미지 세트에 대한 메타데이터를 가져오는 `get-image-set-metadata` 코드 예제입니다.

참고: `outfile`은 필수 파라미터입니다.

```
aws medical-imaging get-image-set-metadata \
    --datastore-id 12345678901234567890123456789012 \
    --image-set-id ea92b0d8838c72a3f25d00d13616f87e \
    --version-id 1 \
    studymetadata.json.gz
```

반환된 메타데이터는 gzip으로 압축되어 studymetadata.json.gz 파일에 저장됩니다. 반환된 JSON 객체의 내용을 보려면 먼저 압축을 풀어야 합니다.

출력:

```
{
    "contentType": "application/json",
    "contentEncoding": "gzip"
}
```

자세한 내용은 AWS HealthImaging 개발자 안내서의 이미지 세트 메타데이터 가져오기를 참조하세요.

- 자세한 API 내용은 명령 참조GetImageSetMetadata의 섹션을 참조하세요. AWS CLI

### get-image-set

다음 코드 예시에서는 get-image-set을 사용하는 방법을 보여 줍니다.

AWS CLI

이미지 세트 속성 가져오기

다음은 이미지 세트의 속성을 가져오는 get-image-set 코드 예제입니다.

```
aws medical-imaging get-image-set \
    --datastore-id 12345678901234567890123456789012 \
    --image-set-id 18f88ac7870584f58d56256646b4d92b \
    --version-id 1
```

출력:

```
{
    "versionId": "1",
    "imageSetWorkflowStatus": "COPIED",
    "updatedAt": 1680027253.471,
    "imageSetId": "18f88ac7870584f58d56256646b4d92b",
    "imageSetState": "ACTIVE",
    "createdAt": 1679592510.753,
    "datastoreId": "12345678901234567890123456789012"
}
```

자세한 내용은 AWS HealthImaging 개발자 안내서의 이미지 세트 속성 가져오기를 참조하세요.

- 자세한 API 내용은 명령 참조GetImageSet의 섹션을 참조하세요. AWS CLI

### list-datastores

다음 코드 예시에서는 list-datastores을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 스토어 나열

다음은 사용 가능한 데이터 스토어를 나열하는 list-datastores 코드 예제입니다.

```
aws medical-imaging list-datastores
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS HealthImaging 개발자 안내서의 데이터 스토어 나열을 참조하세요.

- 자세한 API 내용은 명령 참조ListDatastores의 섹션을 참조하세요. AWS CLI

### `list-dicom-import-jobs`

다음 코드 예시에서는 `list-dicom-import-jobs`을 사용하는 방법을 보여 줍니다.

AWS CLI

dicom 가져오기 작업 나열

다음은 dicom 가져오기 작업을 나열하는 `list-dicom-import-jobs` 코드 예제입니다.

```
aws medical-imaging list-dicom-import-jobs \
    --datastore-id "12345678901234567890123456789012"
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS HealthImaging 개발자 안내서의 가져오기 작업 나열을 참조하세요.

- API 자세한 내용은 AWS CLI 명령 참조의 L istDICOMImport작업을 참조하세요.

### `list-image-set-versions`

다음 코드 예시에서는 `list-image-set-versions`을 사용하는 방법을 보여 줍니다.

AWS CLI

이미지 세트 버전 나열

다음은 이미지 세트의 버전 기록을 나열하는 `list-image-set-versions` 코드 예제입니다.

```
aws medical-imaging list-image-set-versions \
    --datastore-id 12345678901234567890123456789012 \
    --image-set-id ea92b0d8838c72a3f25d00d13616f87e
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 AWS HealthImaging 개발자 안내서의 이미지 세트 버전 나열을 참조하세요.

- 자세한 API 내용은 명령 참조ListImageSetVersions의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 데이터 스토어에 대한 리소스 태그 나열

다음은 데이터 스토어에 대한 태그를 나열하는 `list-tags-for-resource` 코드 예제입니다.

```
aws medical-imaging list-tags-for-resource \
```

```
    --resource-arn "arn:aws:medical-imaging:us-
east-1:123456789012:datastore/12345678901234567890123456789012"
```

출력:

```
{
    "tags":{
        "Deployment":"Development"
    }
}
```

예제 2: 이미지 세트에 대한 리소스 태그 나열

다음은 이미지 세트에 대한 태그를 나열하는 `list-tags-for-resource` 코드 예제입니다.

```
aws medical-imaging list-tags-for-resource \
    --resource-arn "arn:aws:medical-imaging:us-
east-1:123456789012:datastore/12345678901234567890123456789012/
imageset/18f88ac7870584f58d56256646b4d92b"
```

출력:

```
{
    "tags":{
        "Deployment":"Development"
    }
}
```

자세한 내용은 AWS HealthImaging 개발자 안내서의 를 사용하여 리소스 태그 지정 AWS HealthImaging을 참조하세요.

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### `search-image-sets`

다음 코드 예시에서는 `search-image-sets`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: EQUAL 연산자로 이미지 세트를 검색하려면

다음 search-image-sets 코드 예제에서는 EQUAL 연산자를 사용하여 특정 값을 기반으로 이미지 세트를 검색합니다.

```
aws medical-imaging search-image-sets \
    --datastore-id 12345678901234567890123456789012 \
    --search-criteria file://search-criteria.json
```

search-criteria.json의 콘텐츠

```
{
    "filters": [{
        "values": [{"DICOMPatientId" : "SUBJECT08701"}],
        "operator": "EQUAL"
    }]
}
```

출력:

```
{
    "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"
    }]
}
```

예제 2: DICOMStudyDate 및 를 사용하여 BETWEEN 연산자로 이미지 세트를 검색하려면 DICOMStudyTime

다음 search-image-sets 코드 예제는 1990년 1월 1일(오전 12:00)에서 2023년 1월 1일(오전 12:00) 사이에 생성된 DICOM 연구를 사용하여 이미지 세트를 검색합니다.

참고: DICOMStudyTime는 선택 사항입니다. 해당 날짜가 없는 경우 필터링에 제공되는 날짜의 시간 값은 오전 12시(하루의 시작)입니다.

```
aws medical-imaging search-image-sets \
    --datastore-id 12345678901234567890123456789012 \
    --search-criteria file://search-criteria.json
```

search-criteria.json의 콘텐츠

```
{
    "filters": [{
        "values": [{
            "DICOMStudyDateAndTime": {
                "DICOMStudyDate": "19900101",
                "DICOMStudyTime": "000000"
            }
        },
        {
            "DICOMStudyDateAndTime": {
                "DICOMStudyDate": "20230101",
                "DICOMStudyTime": "000000"
            }
        }],
        "operator": "BETWEEN"
    }]
}
```

출력:

```
{
    "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"
    }]
}
```

예제 3: 를 사용하여 BETWEEN 연산자로 이미지 세트를 검색하려면 createdAt (이전에 연구가 지속된 시간)

다음 `search-image-sets` 코드 예제는 UTC 표준 시간대의 시간 범위 HealthImaging 사이에 DICOM 연구가 지속되는 이미지 세트를 검색합니다.

참고: 예제 형식('1985-04-12T23:20:50.52Z') createdAt 으로 제공합니다.

```
aws medical-imaging search-image-sets \
    --datastore-id 12345678901234567890123456789012 \
    --search-criteria  file://search-criteria.json
```

`search-criteria.json`의 콘텐츠

```
{
    "filters": [{
        "values": [{
            "createdAt": "1985-04-12T23:20:50.52Z"
        },
        {
            "createdAt": "2022-04-12T23:20:50.52Z"
        }],
        "operator": "BETWEEN"
    }]
}
```

출력:

```
{
    "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"
    }]
}
```

예제 4: EQUAL 연산자가 켜DICOMSeriesInstanceUID진 상태에서 이미지 세트를 검색BETWEEN updatedAt 하고 updatedAt 필드ASC별로 응답을 정렬하는 방법

다음 `search-image-sets` 코드 예제에서는 EQUAL 연산자가 켜DICOMSeriesInstanceUID진 상태에서 이미지 세트를 검색BETWEEN updatedAt 하고 updatedAt 필드에 ASC 따라 응답을 정렬합니다.

참고: 예제 형식('1985-04-12T23:20:50.52Z') updatedAt 으로 제공합니다.

```
aws medical-imaging search-image-sets \
    --datastore-id 12345678901234567890123456789012 \
    --search-criteria  file://search-criteria.json
```

`search-criteria.json`의 콘텐츠

```
{
    "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"
    }
}
```

출력:

```
{
    "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"
    }]
}
```

자세한 내용은 AWS HealthImaging 개발자 안내서의 이미지 세트 검색을 참조하세요.

- 자세한 API 내용은 명령 참조SearchImageSets의 섹션을 참조하세요. AWS CLI

### `start-dicom-import-job`

다음 코드 예시에서는 `start-dicom-import-job`을 사용하는 방법을 보여 줍니다.

AWS CLI

dicom 가져오기 작업 시작

다음은 dicom 가져오기 작업을 시작하는 `start-dicom-import-job` 코드 예제입니다.

```
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"
```

출력:

```
{
    "datastoreId": "12345678901234567890123456789012",
    "jobId": "09876543210987654321098765432109",
    "jobStatus": "SUBMITTED",
    "submittedAt": "2022-08-12T11:28:11.152000+00:00"
}
```

자세한 내용은 AWS HealthImaging 개발자 안내서의 가져오기 작업 시작을 참조하세요.

- API 자세한 내용은 명령 참조의 S tartDICOMImport작업을 참조하세요. AWS CLI

### `tag-resource`

다음 코드 예시에서는 `tag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 데이터 스토어에 태그 지정

다음은 데이터 스토어에 태그를 지정하는 `tag-resource` 코드 예제입니다.

```
aws medical-imaging tag-resource \
  --resource-arn "arn:aws:medical-imaging:us-
east-1:123456789012:datastore/12345678901234567890123456789012" \
  --tags '{"Deployment":"Development"}'
```

이 명령은 출력을 생성하지 않습니다.

예제 2: 이미지 세트에 태그 지정

다음은 이미지 세트에 태그를 지정하는 tag-resource 코드 예제입니다.

```
aws medical-imaging tag-resource \
    --resource-arn "arn:aws:medical-imaging:us-
east-1:123456789012:datastore/12345678901234567890123456789012/
imageset/18f88ac7870584f58d56256646b4d92b" \
    --tags '{"Deployment":"Development"}'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS HealthImaging 개발자 안내서의 를 사용하여 리소스 태그 지정 AWS HealthImaging을 참조하세요.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### untag-resource

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 데이터 스토어의 태그 해제

다음은 데이터 스토어의 태그를 해제하는 untag-resource 코드 예제입니다.

```
aws medical-imaging untag-resource \
    --resource-arn "arn:aws:medical-imaging:us-
east-1:123456789012:datastore/12345678901234567890123456789012" \
    --tag-keys '["Deployment"]'
```

이 명령은 출력을 생성하지 않습니다.

예제 2: 이미지 세트의 태그 해제

다음은 이미지 세트의 태그를 해제하는 untag-resource 코드 예제입니다.

```
aws medical-imaging untag-resource \
    --resource-arn "arn:aws:medical-imaging:us-
east-1:123456789012:datastore/12345678901234567890123456789012/
imageset/18f88ac7870584f58d56256646b4d92b" \
```

```
    --tag-keys '["Deployment"]'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS HealthImaging 개발자 안내서의 를 사용하여 리소스 태그 지정 AWS HealthImaging을 참조하세요.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-image-set-metadata

다음 코드 예시에서는 update-image-set-metadata을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 이미지 세트 메타데이터에 속성을 삽입하거나 업데이트하려면

다음 update-image-set-metadata 예제에서는 이미지 세트 메타데이터에 속성을 삽입하거나 업데이트합니다.

```
aws medical-imaging update-image-set-metadata \
    --datastore-id 12345678901234567890123456789012 \
    --image-set-id ea92b0d8838c72a3f25d00d13616f87e \
    --latest-version-id 1 \
    --cli-binary-format raw-in-base64-out \
    --update-image-set-metadata-updates file://metadata-updates.json
```

metadata-updates.json의 콘텐츠

```
{
    "DICOMUpdates": {
        "updatableAttributes": "{\"SchemaVersion\":1.1,\"Patient\":{\"DICOM\":
{\"PatientName\":\"MX^MX\"}}}"
    }
}
```

출력:

```
{
    "latestVersionId": "2",
    "imageSetWorkflowStatus": "UPDATING",
```

```
    "updatedAt": 1680042257.908,
    "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
    "imageSetState": "LOCKED",
    "createdAt": 1680027126.436,
    "datastoreId": "12345678901234567890123456789012"
}
```

예제 2: 이미지 세트 메타데이터에서 속성을 제거하려면

다음 update-image-set-metadata 예제에서는 이미지 세트 메타데이터에서 속성을 제거합니다.

```
aws medical-imaging update-image-set-metadata \
    --datastore-id 12345678901234567890123456789012 \
    --image-set-id ea92b0d8838c72a3f25d00d13616f87e \
    --latest-version-id 1 \
    --cli-binary-format raw-in-base64-out \
    --update-image-set-metadata-updates file://metadata-updates.json
```

metadata-updates.json의 콘텐츠

```
{
    "DICOMUpdates": {
        "removableAttributes": "{\"SchemaVersion\":1.1,\"Study\":{\"DICOM\":
{\"StudyDescription\":\"CHEST\"}}}"
    }
}
```

출력:

```
{
    "latestVersionId": "2",
    "imageSetWorkflowStatus": "UPDATING",
    "updatedAt": 1680042257.908,
    "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
    "imageSetState": "LOCKED",
    "createdAt": 1680027126.436,
    "datastoreId": "12345678901234567890123456789012"
}
```

예제 3: 이미지 세트 메타데이터에서 인스턴스를 제거하려면

다음 update-image-set-metadata 예제에서는 이미지 세트 메타데이터에서 인스턴스를 제거합니다.

```
aws medical-imaging update-image-set-metadata \
    --datastore-id 12345678901234567890123456789012 \
    --image-set-id ea92b0d8838c72a3f25d00d13616f87e \
    --latest-version-id 1 \
    --cli-binary-format raw-in-base64-out \
    --update-image-set-metadata-updates file://metadata-updates.json
```

metadata-updates.json의 콘텐츠

```
{
    "DICOMUpdates": {
        "removableAttributes": "{\"SchemaVersion\": 1.1,\"Study\": {\"Series
\": {\"1.1.1.1.1.1.12345.123456789012.123.12345678901234.1\": {\"Instances\":
 {\"1.1.1.1.1.1.12345.123456789012.123.12345678901234.1\": {}}}}}"
    }
}
```

출력:

```
{
    "latestVersionId": "2",
    "imageSetWorkflowStatus": "UPDATING",
    "updatedAt": 1680042257.908,
    "imageSetId": "ea92b0d8838c72a3f25d00d13616f87e",
    "imageSetState": "LOCKED",
    "createdAt": 1680027126.436,
    "datastoreId": "12345678901234567890123456789012"
}
```

예제 4: 이미지 세트를 이전 버전으로 되돌리려면

다음 update-image-set-metadata 예제에서는 이미지 세트를 이전 버전으로 되돌리는 방법을 보여줍니다. CopyImageSet 및 UpdateImageSetMetadata 작업은 이미지 세트의 새 버전을 생성합니다.

```
aws medical-imaging update-image-set-metadata \
    --datastore-id 12345678901234567890123456789012 \
    --image-set-id 53d5fdb05ca4d46ac7ca64b06545c66e \
```

```
    --latest-version-id 3 \
    --cli-binary-format raw-in-base64-out \
    --update-image-set-metadata-updates '{"revertToVersionId": "1"}'
```

출력:

```
{
    "datastoreId": "12345678901234567890123456789012",
    "imageSetId": "53d5fdb05ca4d46ac7ca64b06545c66e",
    "latestVersionId": "4",
    "imageSetState": "LOCKED",
    "imageSetWorkflowStatus": "UPDATING",
    "createdAt": 1680027126.436,
    "updatedAt": 1680042257.908
}
```

예제 5: 인스턴스에 프라이빗 DICOM 데이터 요소를 추가하려면

다음 update-image-set-metadata 예제에서는 이미지 세트 내의 지정된 인스턴스에 프라이빗 요소를 추가하는 방법을 보여줍니다. DICOM 표준은 표준 데이터 요소에 포함될 수 없는 정보를 통신하기 위한 프라이빗 데이터 요소를 허용합니다. UpdateImageSetMetadata 작업을 사용하여 프라이빗 데이터 요소를 생성, 업데이트 및 삭제할 수 있습니다.

```
aws medical-imaging update-image-set-metadata \
    --datastore-id 12345678901234567890123456789012 \
    --image-set-id 53d5fdb05ca4d46ac7ca64b06545c66e \
    --latest-version-id 1 \
    --cli-binary-format raw-in-base64-out \
    --force \
    --update-image-set-metadata-updates file://metadata-updates.json
```

metadata-updates.json의 콘텐츠

```
{
    "DICOMUpdates": {
        "updatableAttributes": "{\"SchemaVersion\": 1.1,\"Study\": {\"Series
\": {\"1.1.1.1.1.1.12345.123456789012.123.12345678901234.1\": {\"Instances
\": {\"1.1.1.1.1.1.12345.123456789012.123.12345678901234.1\": {\"DICOM\":
 {\"001910F9\": \"97\"},\"DICOMVRs\": {\"001910F9\": \"DS\"}}}}}}"
    }
}
```

출력:

```
{
    "latestVersionId": "2",
    "imageSetWorkflowStatus": "UPDATING",
    "updatedAt": 1680042257.908,
    "imageSetId": "53d5fdb05ca4d46ac7ca64b06545c66e",
    "imageSetState": "LOCKED",
    "createdAt": 1680027126.436,
    "datastoreId": "12345678901234567890123456789012"
}
```

예제 6: 프라이빗 DICOM 데이터 요소를 인스턴스로 업데이트하는 방법

다음 update-image-set-metadata 예제에서는 이미지 세트 내의 인스턴스에 속하는 프라이빗 데이터 요소의 값을 업데이트하는 방법을 보여줍니다.

```
aws medical-imaging update-image-set-metadata \
    --datastore-id 12345678901234567890123456789012 \
    --image-set-id 53d5fdb05ca4d46ac7ca64b06545c66e \
    --latest-version-id 1 \
    --cli-binary-format raw-in-base64-out \
    --force \
    --update-image-set-metadata-updates file://metadata-updates.json
```

metadata-updates.json의 콘텐츠

```
{
    "DICOMUpdates": {
        "updatableAttributes": "{\"SchemaVersion\": 1.1,\"Study\": {\"Series
\": {\"1.1.1.1.1.1.12345.123456789012.123.12345678901234.1\": {\"Instances
\": {\"1.1.1.1.1.1.12345.123456789012.123.12345678901234.1\": {\"DICOM\":
 {\"00091001\": \"GE_GENESIS_DD\"}}}}}}"
    }
}
```

출력:

```
{
    "latestVersionId": "2",
    "imageSetWorkflowStatus": "UPDATING",
```

```
    "updatedAt": 1680042257.908,
    "imageSetId": "53d5fdb05ca4d46ac7ca64b06545c66e",
    "imageSetState": "LOCKED",
    "createdAt": 1680027126.436,
    "datastoreId": "12345678901234567890123456789012"
}
```

예제 7: 강제 파라미터SOPInstanceUID로 를 업데이트하려면

다음 `update-image-set-metadata` 예제에서는 강제 파라미터를 SOPInstanceUID사용하여 DICOM 메타데이터 제약 조건을 재정의하여 를 업데이트하는 방법을 보여줍니다.

```
aws medical-imaging update-image-set-metadata \
        --datastore-id 12345678901234567890123456789012 \
        --image-set-id 53d5fdb05ca4d46ac7ca64b06545c66e \
        --latest-version-id 1 \
        --cli-binary-format raw-in-base64-out \
        --force \
        --update-image-set-metadata-updates file://metadata-updates.json
```

`metadata-updates.json`의 콘텐츠

```
{
    "DICOMUpdates": {
        "updatableAttributes": "{\"SchemaVersion\":1.1,\"Study\":{\"Series\":
{\"1.3.6.1.4.1.5962.99.1.3633258862.2104868982.1369432891697.3656.0\":{\"Instances
\":{\"1.3.6.1.4.1.5962.99.1.3633258862.2104868982.1369432891697.3659.0\":{\"DICOM\":
{\"SOPInstanceUID\":
\"1.3.6.1.4.1.5962.99.1.3633258862.2104868982.1369432891697.3659.9\"}}}}}}"
    }
}
```

출력:

```
{
    "latestVersionId": "2",
    "imageSetWorkflowStatus": "UPDATING",
    "updatedAt": 1680042257.908,
    "imageSetId": "53d5fdb05ca4d46ac7ca64b06545c66e",
    "imageSetState": "LOCKED",
    "createdAt": 1680027126.436,
    "datastoreId": "12345678901234567890123456789012"
```

```
}
```

자세한 내용은 AWS HealthImaging 개발자 안내서의 이미지 세트 메타데이터 업데이트를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateImageSetMetadata의 섹션을 참조하세요. AWS CLI

## HealthLake 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 HealthLake.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `create-fhir-datastore`

다음 코드 예시에서는 `create-fhir-datastore`을 사용하는 방법을 보여 줍니다.

AWS CLI

FHIR 데이터 스토어를 생성합니다.

다음 `create-fhir-datastore` 예제에서는 Amazon 에서 새 데이터 스토어를 생성하는 방법을 보여줍니다 HealthLake.

```
aws healthlake create-fhir-datastore \
    --region us-east-1 \
    --datastore-type-version R4 \
    --datastore-type-version R4 \
    --datastore-name "FhirTestDatastore"
```

출력:

```
{
    "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)"
}
```

자세한 내용은 Amazon HealthLake 개발자 안내서의 FHIR 데이터 스토어 생성 및 모니터링을 참조하세요.

- 자세한 API 내용은 명령 참조CreateFhirDatastore의 섹션을 참조하세요. AWS CLI

### `delete-fhir-datastore`

다음 코드 예시에서는 `delete-fhir-datastore`을 사용하는 방법을 보여 줍니다.

AWS CLI

FHIR 데이터 스토어를 삭제하려면

다음 `delete-fhir-datastore` 예제에서는 Amazon 에서 데이터 스토어와 해당 콘텐츠를 모두 삭제하는 방법을 보여줍니다 HealthLake.

```
aws healthlake delete-fhir-datastore \
    --datastore-id (Data Store ID) \
    --region us-east-1
```

출력:

```
{
    "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)"
}
```

자세한 내용은 Amazon HealthLake 개발자 안내서의 <https://docs.aws.amazon.com/healthlake/latest/devguide/working-with-FHIR-healthlake.html> FHIR 데이터 스토어 생성 및 모니터링을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteFhirDatastore의 섹션을 참조하세요. AWS CLI

### `describe-fhir-datastore`

다음 코드 예시에서는 `describe-fhir-datastore`을 사용하는 방법을 보여 줍니다.

AWS CLI

FHIR 데이터 스토어를 설명하려면

다음 `describe-fhir-datastore` 예제에서는 Amazon 에서 데이터 스토어의 속성을 찾는 방법을 보여줍니다 HealthLake.

```
aws healthlake describe-fhir-datastore \
    --datastore-id "1f2f459836ac6c513ce899f9e4f66a59" \
    --region us-east-1
```

출력:

```
{
    "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)"
    }
}
```

자세한 내용은 Amazon HealthLake 개발자 안내서의 FHIR 데이터 스토어 생성 및 모니터링을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeFhirDatastore의 섹션을 참조하세요. AWS CLI

### describe-fhir-export-job

다음 코드 예시에서는 describe-fhir-export-job을 사용하는 방법을 보여 줍니다.

AWS CLI

FHIR 내보내기 작업을 설명하려면

다음 describe-fhir-export-job 예제는 Amazon 에서 FHIR 내보내기 작업의 속성을 찾는 방법을 보여줍니다 HealthLake.

```
aws healthlake describe-fhir-export-job \
    --datastore-id (Datastore ID) \
    --job-id 9b9a51943afaedd0a8c0c26c49135a31
```

출력:

```
{
    "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)"
    }
}
```

자세한 내용은 Amazon HealthLake 개발자 안내서의 FHIR 데이터 스토어에서 파일 내보내기를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeFhirExportJob의 섹션을 참조하세요. AWS CLI

### describe-fhir-import-job

다음 코드 예시에서는 describe-fhir-import-job을 사용하는 방법을 보여 줍니다.

AWS CLI

FHIR 가져오기 작업을 설명하려면

다음 `describe-fhir-import-job` 예제에서는 Amazon 를 사용하여 FHIR 가져오기 작업의 속성을 학습하는 방법을 보여줍니다 HealthLake.

```
aws healthlake describe-fhir-import-job \
    --datastore-id (Datastore ID) \
    --job-id c145fbb27b192af392f8ce6e7838e34f \
    --region us-east-1
```

출력:

```
{
    "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)"
    }
}
```

자세한 내용은 Amazon HealthLake 개발자 안내서의 FHIR 데이터 스토어로 파일 가져오기를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeFhirImportJob의 섹션을 참조하세요. AWS CLI

### `list-fhir-datastores`

다음 코드 예시에서는 `list-fhir-datastores`을 사용하는 방법을 보여 줍니다.

AWS CLI

FHIR 데이터 스토어를 나열하려면

다음 `list-fhir-datastores` 예제에서는 명령을 사용하는 방법과 사용자가 Amazon 의 Data Store 상태를 기반으로 결과를 필터링하는 방법을 보여줍니다 HealthLake.

```
aws healthlake list-fhir-datastores \
    --region us-east-1 \
    --filter DatastoreStatus=ACTIVE
```

출력:

```
{
    "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>"
    }
    ]
}
```

자세한 내용은 Amazon HealthLake 개발자 안내서의 FHIR 데이터 스토어 생성 및 모니터링을 참조하세요.

- 자세한 API 내용은 명령 참조ListFhirDatastores의 섹션을 참조하세요. AWS CLI

### list-fhir-export-jobs

다음 코드 예시에서는 list-fhir-export-jobs을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 FHIR 내보내기 작업을 나열하려면

다음 list-fhir-export-jobs 예제에서는 명령을 사용하여 계정과 연결된 내보내기 작업 목록을 보는 방법을 보여줍니다.

```
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)
```

출력:

```
{
    "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
```

자세한 내용은 Amazon HealthLake 개발자 안내서의 FHIR 데이터 스토어에서 파일 내보내기를 참조하세요.

- 자세한 API 내용은 명령 참조ListFhirExportJobs의 섹션을 참조하세요. AWS CLI

### `list-fhir-import-jobs`

다음 코드 예시에서는 `list-fhir-import-jobs`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 FHIR 가져오기 작업을 나열하려면

다음 `list-fhir-import-jobs` 예제에서는 명령을 사용하여 계정과 연결된 모든 가져오기 작업 목록을 보는 방법을 보여줍니다.

```
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)
```

출력:

```
{
    "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
```

자세한 내용은 Amazon HealthLake 개발자 안내서의 FHIR Data Store로 파일 가져오기를 참조하세요.

- 자세한 API 내용은 명령 참조ListFhirImportJobs의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 스토어의 태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 지정된 데이터 스토어와 연결된 태그를 나열합니다.

```
aws healthlake list-tags-for-resource \
    --resource-arn "arn:aws:healthlake:us-east-1:674914422125:datastore/
fhir/0725c83f4307f263e16fd56b6d8ebdbe" \
    --region us-east-1
```

출력:

```
{
    "tags": {
        "key": "value",
        "key1": "value1"
    }
}
```

자세한 내용은 Amazon 개발자 안내서의 Amazon에서 리소스 태그 지정 HealthLake을 참조하세요. HealthLake

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### `start-fhir-export-job`

다음 코드 예시에서는 `start-fhir-export-job`을 사용하는 방법을 보여 줍니다.

AWS CLI

FHIR 내보내기 작업을 시작하려면

다음 `start-fhir-export-job` 예제에서는 Amazon 를 사용하여 FHIR 내보내기 작업을 시작하는 방법을 보여줍니다 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)
```

출력:

```
{
    "DatastoreId": "(Datastore ID)",
    "JobStatus": "SUBMITTED",
    "JobId": "9b9a51943afaedd0a8c0c26c49135a31"
}
```

자세한 내용은 Amazon HealthLake 개발자 안내서의 FHIR 데이터 스토어에서 파일 내보내기를 참조하세요.

- 자세한 API 내용은 명령 참조StartFhirExportJob의 섹션을 참조하세요. AWS CLI

### `start-fhir-import-job`

다음 코드 예시에서는 `start-fhir-import-job`을 사용하는 방법을 보여 줍니다.

AWS CLI

FHIR 가져오기 작업을 시작하려면

다음 `start-fhir-import-job` 예제에서는 Amazon 를 사용하여 FHIR 가져오기 작업을 시작하는 방법을 보여줍니다 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
```

출력:

```
{
```

```
    "DatastoreId": "(Datastore ID)",
    "JobStatus": "SUBMITTED",
    "JobId": "c145fbb27b192af392f8ce6e7838e34f"
}
```

자세한 내용은 Amazon HeatlhLake 개발자 안내서의 FHIR Data Store 'https://docs.aws.amazon.com/healthlake/latest/devguide/import-datastore.html로 파일 가져오기를 참조하세요.

- 자세한 API 내용은 명령 참조StartFhirImportJob의 섹션을 참조하세요. AWS CLI

### `tag-resource`

다음 코드 예시에서는 `tag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

Data Store에 태그를 추가하려면

다음 `tag-resource` 예제에서는 Data Store에 태그를 추가하는 방법을 보여줍니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon 개발자 안내서의 '데이터 스토어에 태그 추가 <https://docs.aws.amazon.com/healthlake/latest/devguide/add-a-tag.html>'__를 참조하세요.  HealthLake

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### `untag-resource`

다음 코드 예시에서는 `untag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 스토어에서 태그를 제거합니다.

다음 `untag-resource` 예제에서는 데이터 스토어에서 태그를 제거하는 방법을 보여줍니다.

```
aws healthlake untag-resource \
    --resource-arn "arn:aws:healthlake:us-east-1:674914422125:datastore/fhir/
b91723d65c6fdeb1d26543a49d2ed1fa" \
    --tag-keys '["key1"]' \
    --region us-east-1
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon HealthLake 개발자 안내서의 데이터 스토어에서 태그 제거를 참조하세요.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

## HealthOmics 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 HealthOmics.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `abort-multipart-read-set-upload`

다음 코드 예시에서는 `abort-multipart-read-set-upload`을 사용하는 방법을 보여 줍니다.

AWS CLI

멀티파트 읽기 세트 업로드를 중지하려면

다음 `abort-multipart-read-set-upload` 예제에서는 HealthOmics 시퀀스 스토어에 멀티파트 읽기 세트 업로드를 중지합니다.

```
aws omics abort-multipart-read-set-upload \
    --sequence-store-id 0123456789 \
    --upload-id 1122334455
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS HealthOmics 사용 설명서의 시퀀스 스토어에 직접 업로드를 참조하세요.

- 자세한 API 내용은 명령 참조AbortMultipartReadSetUpload의 섹션을 참조하세요. AWS CLI

### accept-share

다음 코드 예시에서는 accept-share을 사용하는 방법을 보여 줍니다.

AWS CLI

분석 스토어 데이터의 공유를 수락하려면

다음 accept-share 예제에서는 HealthOmics 분석 스토어 데이터의 공유를 허용합니다.

```
aws omics accept-share \
    ----share-id "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

출력:

```
{
    "status": "ACTIVATING"
}
```

자세한 내용은 AWS HealthOmics 사용 설명서의 교차 계정 공유를 참조하세요.

- 자세한 API 내용은 명령 참조AcceptShare의 섹션을 참조하세요. AWS CLI

### batch-delete-read-set

다음 코드 예시에서는 batch-delete-read-set을 사용하는 방법을 보여 줍니다.

AWS CLI

여러 읽기 세트를 삭제하려면

다음 batch-delete-read-set 예제에서는 두 개의 읽기 세트를 삭제합니다.

```
aws omics batch-delete-read-set \
    --sequence-store-id 1234567890 \
    --ids 1234567890 0123456789
```

지정된 읽기 세트를 삭제하는 동안 오류가 발생하면 서비스가 오류 목록을 반환합니다.

```
{
    "errors": [
        {
            "code": "",
            "id": "0123456789",
            "message": "The specified readset does not exist."
        }
    ]
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 스토리지를 참조하세요.

- 자세한 API 내용은 명령 참조BatchDeleteReadSet의 섹션을 참조하세요. AWS CLI

### cancel-annotation-import-job

다음 코드 예시에서는 cancel-annotation-import-job을 사용하는 방법을 보여 줍니다.

AWS CLI

주석 가져오기 작업을 취소하려면

다음 cancel-annotation-import-job 예제에서는 ID 를 사용하여 주석 가져오기 작업을 취소합니다04f57618-xmpl-4fd0-9349-e5a85aefb997.

```
aws omics cancel-annotation-import-job \
    --job-id 04f57618-xmpl-4fd0-9349-e5a85aefb997
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics Analytics를 참조하세요.

- 자세한 API 내용은 명령 참조CancelAnnotationImportJob의 섹션을 참조하세요. AWS CLI

### cancel-run

다음 코드 예시에서는 cancel-run을 사용하는 방법을 보여 줍니다.

AWS CLI

실행을 취소하려면

다음 cancel-run 예제에서는 ID 를 사용하여 실행을 취소합니다1234567.

```
aws omics cancel-run \
    --id 1234567
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 워크플로를 참조하세요.

- 자세한 API 내용은 명령 참조CancelRun의 섹션을 참조하세요. AWS CLI

### cancel-variant-import-job

다음 코드 예시에서는 cancel-variant-import-job을 사용하는 방법을 보여 줍니다.

AWS CLI

변형 가져오기 작업을 취소하려면

다음 cancel-variant-import-job 예제에서는 ID 를 사용하여 변형 가져오기 작업을 취소합니다69cb65d6-xmpl-4a4a-9025-4565794b684e.

```
aws omics cancel-variant-import-job \
    --job-id 69cb65d6-xmpl-4a4a-9025-4565794b684e
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics Analytics를 참조하세요.

- 자세한 API 내용은 명령 참조CancelVariantImportJob의 섹션을 참조하세요. AWS CLI

### complete-multipart-read-set-upload

다음 코드 예시에서는 complete-multipart-read-set-upload을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 구성 요소를 업로드한 후 멀티파트 업로드를 종료하려면

다음 complete-multipart-read-set-upload 예제에서는 모든 구성 요소가 업로드되면 시퀀스 스토어에 멀티파트 업로드를 완료합니다.

```
aws omics complete-multipart-read-set-upload \
```

```
    --sequence-store-id 0123456789 \
    --upload-id 1122334455 \
    --parts '[{"checksum":"gaCBQMe+rpCFZxLpoP6gydBoXaKKDA/
Vobh5zBDb4W4=","partNumber":1,"partSource":"SOURCE1"}]'
```

출력:

```
{
    "readSetId": "0000000001"
    "readSetId": "0000000002"
    "readSetId": "0000000003"
}
```

자세한 내용은 AWS HealthOmics 사용 설명서의 시퀀스 스토어에 직접 업로드를 참조하세요.

- 자세한 API 내용은 명령 참조CompleteMultipartReadSetUpload의 섹션을 참조하세요. AWS CLI

### create-annotation-store-version

다음 코드 예시에서는 create-annotation-store-version을 사용하는 방법을 보여 줍니다.

AWS CLI

주석 스토어의 새 버전을 생성하려면

다음 create-annotation-store-version 예제에서는 새 버전의 주석 저장소를 생성합니다.

```
aws omics create-annotation-store-version \
    --name my_annotation_store \
    --version-name my_version
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS HealthOmics 사용 설명서의 새 버전의 주석 스토어 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateAnnotationStoreVersion의 섹션을 참조하세요. AWS CLI

### create-annotation-store

다음 코드 예시에서는 create-annotation-store을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: VCF 주석 저장소 생성

다음 create-annotation-store 예제에서는 VCF 형식 주석 저장소를 생성합니다.

```
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
```

출력:

```
{
    "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"
}
```

예제 2: TSV 주석 저장소 생성

다음 create-annotation-store 예제에서는 TSV 형식 주석 저장소를 생성합니다.

```
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 는 주석의 형식 옵션을 구성합니다.

```
{
    "tsvStoreOptions": {
        "annotationType": "CHR_START_END_ZERO_BASE",
        "formatToHeader": {
            "CHR": "chromosome",
            "START": "start",
            "END": "end"
        },
        "schema": [
            {
                "chromosome": "STRING"
            },
            {
                "start": "LONG"
            },
            {
                "end": "LONG"
            },
            {
                "name": "STRING"
            }
        ]
    }
}
```

출력:

```
{
    "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"
                }
            ]
        }
    }
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics Analytics를 참조하세요.

- 자세한 API 내용은 명령 참조CreateAnnotationStore의 섹션을 참조하세요. AWS CLI

### create-multipart-read-set-upload

다음 코드 예시에서는 create-multipart-read-set-upload을 사용하는 방법을 보여 줍니다.

AWS CLI

멀티파트 읽기 세트 업로드를 시작하려면

다음 create-multipart-read-set-upload 예제에서는 멀티파트 읽기 세트 업로드를 시작합니다.

```
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"
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS HealthOmics 사용 설명서의 시퀀스 스토어에 직접 업로드를 참조하세요.

- 자세한 API 내용은 명령 참조CreateMultipartReadSetUpload의 섹션을 참조하세요. AWS CLI

### create-reference-store

다음 코드 예시에서는 create-reference-store을 사용하는 방법을 보여 줍니다.

AWS CLI

참조 스토어를 생성하려면

다음 create-reference-store 예제에서는 참조 스토어 를 생성합니다my-ref-store.

```
aws omics create-reference-store \
    --name my-ref-store
```

출력:

```
{
    "arn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890",
```

```
    "creationTime": "2022-11-22T22:13:25.947Z",
    "id": "1234567890",
    "name": "my-ref-store"
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 스토리지를 참조하세요.

- 자세한 API 내용은 명령 참조CreateReferenceStore의 섹션을 참조하세요. AWS CLI

### create-run-group

다음 코드 예시에서는 create-run-group을 사용하는 방법을 보여 줍니다.

AWS CLI

실행 그룹을 생성하려면

다음 create-run-group 예제에서는 라는 실행 그룹을 생성합니다cram-converter.

```
aws omics create-run-group \
    --name cram-converter \
    --max-cpus 20 \
    --max-duration 600
```

출력:

```
{
    "arn": "arn:aws:omics:us-west-2:123456789012:runGroup/1234567",
    "id": "1234567",
    "tags": {}
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 워크플로를 참조하세요.

- 자세한 API 내용은 명령 참조CreateRunGroup의 섹션을 참조하세요. AWS CLI

### create-sequence-store

다음 코드 예시에서는 create-sequence-store을 사용하는 방법을 보여 줍니다.

AWS CLI

시퀀스 스토어를 생성하려면

다음 create-sequence-store 예제에서는 시퀀스 스토어를 생성합니다.

```
aws omics create-sequence-store \
    --name my-seq-store
```

출력:

```
{
    "arn": "arn:aws:omics:us-west-2:123456789012:sequenceStore/1234567890",
    "creationTime": "2022-11-23T01:24:33.629Z",
    "id": "1234567890",
    "name": "my-seq-store"
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 스토리지를 참조하세요.

- 자세한 API 내용은 명령 참조CreateSequenceStore의 섹션을 참조하세요. AWS CLI

### create-share

다음 코드 예시에서는 create-share을 사용하는 방법을 보여 줍니다.

AWS CLI

HealthOmics 분석 스토어의 공유를 생성하려면

다음 create-share 예제에서는 계정 외부의 구독자가 수락할 수 있는 HealthOmics 분석 스토어의 공유를 생성하는 방법을 보여줍니다.

```
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"
```

출력:

```
{
    "shareId": "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a",
    "name": "my_Share-123",
    "status": "PENDING"
}
```

자세한 내용은 AWS HealthOmics 사용 설명서의 교차 계정 공유를 참조하세요.

- 자세한 API 내용은 명령 참조CreateShare의 섹션을 참조하세요. AWS CLI

### create-variant-store

다음 코드 예시에서는 create-variant-store을 사용하는 방법을 보여 줍니다.

AWS CLI

변형 스토어를 생성하려면

다음 create-variant-store 예제에서는 라는 변형 스토어를 생성합니다my_var_store.

```
aws omics create-variant-store \
    --name my_var_store \
    --reference referenceArn=arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics Analytics를 참조하세요.

- 자세한 API 내용은 명령 참조CreateVariantStore의 섹션을 참조하세요. AWS CLI

### create-workflow

다음 코드 예시에서는 create-workflow을 사용하는 방법을 보여 줍니다.

AWS CLI

워크플로를 생성하려면

다음 create-workflow 예제에서는 WDL 워크플로를 생성합니다.

```
aws omics create-workflow \
    --name cram-converter \
    --engine WDL \
    --definition-zip fileb://workflow-crambam.zip \
    --parameter-template file://workflow-params.json
```

workflow-crambam.zip 는 워크플로 정의를 포함하는 ZIP 아카이브입니다. 는 워크플로의 런타임 파라미터를 workflow-params.json 정의합니다.

```
{
    "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
    }
}
```

출력:

```
{
    "arn": "arn:aws:omics:us-west-2:123456789012:workflow/1234567",
    "id": "1234567",
    "status": "CREATING",
    "tags": {}
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 워크플로를 참조하세요.

- 자세한 API 내용은 명령 참조CreateWorkflow의 섹션을 참조하세요. AWS CLI

### delete-annotation-store-versions

다음 코드 예시에서는 delete-annotation-store-versions을 사용하는 방법을 보여 줍니다.

AWS CLI

주석 저장소 버전을 삭제하려면

다음 delete-annotation-store-versions 예제에서는 주석 저장소 버전을 삭제합니다.

```
aws omics delete-annotation-store-versions \
    --name my_annotation_store \
    --versions my_version
```

출력:

```
{
    "errors": []
}
```

자세한 내용은 AWS HealthOmics 사용 설명서의 새 버전의 주석 스토어 생성을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteAnnotationStoreVersions의 섹션을 참조하세요. AWS CLI

### delete-annotation-store

다음 코드 예시에서는 delete-annotation-store을 사용하는 방법을 보여 줍니다.

AWS CLI

주석 저장소를 삭제하려면

다음 delete-annotation-store 예제에서는 라는 주석 저장소를 삭제합니다my_vcf_store.

```
aws omics delete-annotation-store \
    --name my_vcf_store
```

출력:

```
{
    "status": "DELETING"
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics Analytics를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteAnnotationStore의 섹션을 참조하세요. AWS CLI

### delete-reference-store

다음 코드 예시에서는 delete-reference-store을 사용하는 방법을 보여 줍니다.

AWS CLI

참조 스토어를 삭제하려면

다음 delete-reference-store 예제에서는 ID가 인 참조 스토어를 삭제합니다1234567890.

```
aws omics delete-reference-store \
    --id 1234567890
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 스토리지를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteReferenceStore의 섹션을 참조하세요. AWS CLI

### delete-reference

다음 코드 예시에서는 delete-reference을 사용하는 방법을 보여 줍니다.

AWS CLI

참조를 삭제하려면

다음 delete-reference 예제에서는 참조를 삭제합니다.

```
aws omics delete-reference \
    --reference-store-id 1234567890 \
    --id 1234567890
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 스토리지를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteReference의 섹션을 참조하세요. AWS CLI

### delete-run-group

다음 코드 예시에서는 delete-run-group을 사용하는 방법을 보여 줍니다.

AWS CLI

실행 그룹을 삭제하려면

다음 delete-run-group 예제에서는 ID가 인 실행 그룹을 삭제합니다1234567.

```
aws omics delete-run-group \
    --id 1234567
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 워크플로를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteRunGroup의 섹션을 참조하세요. AWS CLI

### delete-run

다음 코드 예시에서는 delete-run을 사용하는 방법을 보여 줍니다.

AWS CLI

워크플로 실행을 삭제하려면

다음 delete-run 예제에서는 ID가 인 실행을 삭제합니다1234567.

```
aws omics delete-run \
    --id 1234567
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 워크플로를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteRun의 섹션을 참조하세요. AWS CLI

### delete-sequence-store

다음 코드 예시에서는 delete-sequence-store을 사용하는 방법을 보여 줍니다.

AWS CLI

시퀀스 스토어를 삭제하려면

다음 delete-sequence-store 예제에서는 ID가 인 시퀀스 스토어를 삭제합니다1234567890.

```
aws omics delete-sequence-store \
    --id 1234567890
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 스토리지를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteSequenceStore의 섹션을 참조하세요. AWS CLI

### delete-share

다음 코드 예시에서는 delete-share을 사용하는 방법을 보여 줍니다.

AWS CLI

HealthOmics 분석 데이터의 공유를 삭제하려면

다음 delete-share 예제에서는 분석 데이터의 교차 계정 공유를 삭제합니다.

```
aws omics delete-share \
    --share-id "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

출력:

```
{
    "status": "DELETING"
}
```

자세한 내용은 AWS HealthOmics 사용 설명서의 교차 계정 공유를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteShare의 섹션을 참조하세요. AWS CLI

### delete-variant-store

다음 코드 예시에서는 delete-variant-store을 사용하는 방법을 보여 줍니다.

AWS CLI

변형 스토어를 삭제하려면

다음 delete-variant-store 예제에서는 라는 변형 스토어를 삭제합니다my_var_store.

```
aws omics delete-variant-store \
    --name my_var_store
```

출력:

```
{
    "status": "DELETING"
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 분석을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteVariantStore의 섹션을 참조하세요. AWS CLI

### delete-workflow

다음 코드 예시에서는 delete-workflow을 사용하는 방법을 보여 줍니다.

AWS CLI

워크플로를 삭제하려면

다음 delete-workflow 예제에서는 ID가 인 워크플로를 삭제합니다1234567.

```
aws omics delete-workflow \
    --id 1234567
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 워크플로를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteWorkflow의 섹션을 참조하세요. AWS CLI

### get-annotation-import-job

다음 코드 예시에서는 get-annotation-import-job을 사용하는 방법을 보여 줍니다.

AWS CLI

주석 가져오기 작업을 보려면

다음 get-annotation-import-job 예제에서는 주석 가져오기 작업에 대한 세부 정보를 가져옵니다.

```
aws omics get-annotation-import-job \
    --job-id 984162c7-xmpl-4d23-ab47-286f7950bfbf
```

출력:

```
{
    "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-
W8O1XMPL7QZ",
    "runLeftNormalization": false,
    "status": "COMPLETED",
    "updateTime": "2022-11-30T01:42:39.134009Z"
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics Analytics를 참조하세요.

- 자세한 API 내용은 명령 참조GetAnnotationImportJob의 섹션을 참조하세요. AWS CLI

### get-annotation-store-version

다음 코드 예시에서는 get-annotation-store-version을 사용하는 방법을 보여 줍니다.

AWS CLI

주석 저장소 버전의 메타데이터를 검색하려면

다음 get-annotation-store-version 예제에서는 요청된 주석 저장소 버전의 메타데이터를 검색합니다.

```
aws omics get-annotation-store-version \
    --name my_annotation_store \
    --version-name my_version
```

출력:

```
{
    "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
}
```

자세한 내용은 AWS HealthOmics 사용 설명서의 새 버전의 주석 스토어 생성을 참조하세요.

- 자세한 API 내용은 명령 참조GetAnnotationStoreVersion의 섹션을 참조하세요. AWS CLI

### get-annotation-store

다음 코드 예시에서는 get-annotation-store을 사용하는 방법을 보여 줍니다.

AWS CLI

주석 스토어를 보려면

다음 get-annotation-store 예제에서는 라는 주석 저장소에 대한 세부 정보를 가져옵니다my_ann_store.

```
aws omics get-annotation-store \
    --name my_ann_store
```

출력:

```
{
    "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": {}
```

```
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics Analytics를 참조하세요.

- 자세한 API 내용은 명령 참조GetAnnotationStore의 섹션을 참조하세요. AWS CLI

### get-read-set-activation-job

다음 코드 예시에서는 get-read-set-activation-job을 사용하는 방법을 보여 줍니다.

AWS CLI

읽기 세트 활성화 작업을 보려면

다음 get-read-set-activation-job 예제에서는 읽기 세트 활성화 작업에 대한 세부 정보를 가져옵니다.

```
aws omics get-read-set-activation-job \
    --sequence-store-id 1234567890 \
    --id 1234567890
```

출력:

```
{
    "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."
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 스토리지를 참조하세요.

- 자세한 API 내용은 명령 참조GetReadSetActivationJob의 섹션을 참조하세요. AWS CLI

### get-read-set-export-job

다음 코드 예시에서는 get-read-set-export-job을 사용하는 방법을 보여 줍니다.

AWS CLI

읽기 세트 내보내기 작업을 보려면

다음 get-read-set-export-job 예제에서는 읽기 세트 내보내기 작업에 대한 세부 정보를 가져옵니다.

```
aws omics get-read-set-export-job \
    --sequence-store-id 1234567890 \
    --id 1234567890
```

출력:

```
{
    "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."
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 스토리지를 참조하세요.

- 자세한 API 내용은 명령 참조GetReadSetExportJob의 섹션을 참조하세요. AWS CLI

### get-read-set-import-job

다음 코드 예시에서는 get-read-set-import-job을 사용하는 방법을 보여 줍니다.

AWS CLI

읽기 세트 가져오기 작업을 보려면

다음 get-read-set-import-job 예제에서는 읽기 세트 가져오기 작업에 대한 세부 정보를 가져옵니다.

```
aws omics get-read-set-import-job \
    --sequence-store-id 1234567890 \
    --id 1234567890
```

출력:

```
{
    "creationTime": "2022-11-23T01:36:38.158Z",
    "id": "1234567890",
    "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W8O1XMPL7QZ",
    "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."
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 스토리지를 참조하세요.

- 자세한 API 내용은 명령 참조GetReadSetImportJob의 섹션을 참조하세요. AWS CLI

### `get-read-set-metadata`

다음 코드 예시에서는 `get-read-set-metadata`을 사용하는 방법을 보여 줍니다.

AWS CLI

읽기 세트를 보려면

다음 get-read-set-metadata 예제에서는 읽기 세트의 파일에 대한 세부 정보를 가져옵니다.

```
aws omics get-read-set-metadata \
    --sequence-store-id 1234567890 \
    --id 1234567890
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 스토리지를 참조하세요.

- 자세한 API 내용은 명령 참조GetReadSetMetadata의 섹션을 참조하세요. AWS CLI

### get-read-set

다음 코드 예시에서는 get-read-set을 사용하는 방법을 보여 줍니다.

AWS CLI

읽기 세트를 다운로드하려면

다음 get-read-set 예제에서는 읽기 세트의 파트 3을 로 다운로드합니다1234567890.3.bam.

```
aws omics get-read-set \
    --sequence-store-id 1234567890 \
    --id 1234567890 \
    --part-number 3  1234567890.3.bam
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 스토리지를 참조하세요.

- 자세한 API 내용은 명령 참조GetReadSet의 섹션을 참조하세요. AWS CLI

### get-reference-import-job

다음 코드 예시에서는 get-reference-import-job을 사용하는 방법을 보여 줍니다.

AWS CLI

참조 가져오기 작업을 보려면

다음 get-reference-import-job 예제에서는 참조 가져오기 작업에 대한 세부 정보를 가져옵니다.

```
aws omics get-reference-import-job \
    --reference-store-id 1234567890 \
    --id 1234567890
```

출력:

```
{
    "creationTime": "2022-11-22T22:25:41.124Z",
    "id": "1234567890",
    "referenceStoreId": "1234567890",
    "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W8O1XMPL7QZ",
```

```
    "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."
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 스토리지를 참조하세요.

- 자세한 API 내용은 명령 참조GetReferenceImportJob의 섹션을 참조하세요. AWS CLI

### `get-reference-metadata`

다음 코드 예시에서는 `get-reference-metadata`을 사용하는 방법을 보여 줍니다.

AWS CLI

참조를 보려면

다음 `get-reference-metadata` 예제에서는 참조에 대한 세부 정보를 가져옵니다.

```
aws omics get-reference-metadata \
    --reference-store-id 1234567890 \
    --id 1234567890
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 스토리지를 참조하세요.

- 자세한 API 내용은 명령 참조GetReferenceMetadata의 섹션을 참조하세요. AWS CLI

### get-reference-store

다음 코드 예시에서는 get-reference-store을 사용하는 방법을 보여 줍니다.

AWS CLI

참조 스토어를 보려면

다음 get-reference-store 예제에서는 참조 스토어에 대한 세부 정보를 가져옵니다.

```
aws omics get-reference-store \
    --id 1234567890
```

출력:

```
{
    "arn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890",
    "creationTime": "2022-09-23T23:27:20.364Z",
    "id": "1234567890",
    "name": "my-rstore-0"
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 스토리지를 참조하세요.

- 자세한 API 내용은 명령 참조GetReferenceStore의 섹션을 참조하세요. AWS CLI

### get-reference

다음 코드 예시에서는 get-reference을 사용하는 방법을 보여 줍니다.

AWS CLI

유전체 참조를 다운로드하려면

다음 get-reference 예제에서는 유전체의 파트 1을 로 다운로드합니다hg38.1.fa.

```
aws omics get-reference \
    --reference-store-id 1234567890 \
    --id 1234567890 \
    --part-number 1 hg38.1.fa
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 스토리지를 참조하세요.

- 자세한 API 내용은 명령 참조GetReference의 섹션을 참조하세요. AWS CLI

### get-run-group

다음 코드 예시에서는 get-run-group을 사용하는 방법을 보여 줍니다.

AWS CLI

실행 그룹을 보려면

다음 get-run-group 예제에서는 실행 그룹에 대한 세부 정보를 가져옵니다.

```
aws omics get-run-group \
    --id 1234567
```

출력:

```
{
    "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": {}
```

```
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 워크플로를 참조하세요.

- 자세한 API 내용은 명령 참조GetRunGroup의 섹션을 참조하세요. AWS CLI

### get-run-task

다음 코드 예시에서는 get-run-task을 사용하는 방법을 보여 줍니다.

AWS CLI

작업을 보려면

다음 get-run-task 예제에서는 워크플로 작업에 대한 세부 정보를 가져옵니다.

```
aws omics get-run-task \
    --id 1234567 \
    --task-id 1234567
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 워크플로를 참조하세요.

- 자세한 API 내용은 명령 참조GetRunTask의 섹션을 참조하세요. AWS CLI

### get-run

다음 코드 예시에서는 get-run을 사용하는 방법을 보여 줍니다.

AWS CLI

워크플로 실행을 보려면

다음 get-run 예제에서는 워크플로 실행에 대한 세부 정보를 가져옵니다.

```
aws omics get-run \
    --id 1234567
```

출력:

```
{
    "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:3884c62eb0e53fa92459ed9bff133ae6",
        "s3://omics-artifacts-01d6xmpl4e72dd32/Homo_sapiens_assembly38.fasta":
 "etag:e307d81c605fb91b7720a08f00276842-388",
        "s3://omics-artifacts-01d6xmpl4e72dd32/NA12878.cram":
 "etag:a9f52976381286c6143b5cc681671ec6"
    },
    "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W8O1XMPL7QZ",
    "startedBy": "arn:aws:iam::123456789012:user/laptop-2020",
    "status": "STARTING",
```

```
    "tags": {},
    "workflowId": "1234567",
    "workflowType": "PRIVATE"
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 워크플로를 참조하세요.

- 자세한 API 내용은 명령 참조GetRun의 섹션을 참조하세요. AWS CLI

### get-sequence-store

다음 코드 예시에서는 get-sequence-store을 사용하는 방법을 보여 줍니다.

AWS CLI

시퀀스 스토어를 보려면

다음 get-sequence-store 예제에서는 ID가 인 시퀀스 스토어에 대한 세부 정보를 가져옵니다1234567890.

```
aws omics get-sequence-store \
    --id 1234567890
```

출력:

```
{
    "arn": "arn:aws:omics:us-east-1:123456789012:sequenceStore/1234567890",
    "creationTime": "2022-11-23T19:55:48.376Z",
    "id": "1234567890",
    "name": "my-seq-store"
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 스토리지를 참조하세요.

- 자세한 API 내용은 명령 참조GetSequenceStore의 섹션을 참조하세요. AWS CLI

### get-share

다음 코드 예시에서는 get-share을 사용하는 방법을 보여 줍니다.

AWS CLI

HealthOmics 분석 데이터의 공유에 대한 메타데이터를 검색하려면

다음 get-share 예제에서는 분석 데이터의 교차 계정 공유에 대한 메타데이터를 검색합니다.

```
aws omics get-share \
    --share-id "495c21bedc889d07d0ab69d710a6841e-dd75ab7a1a9c384fa848b5bd8e5a7e0a"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS HealthOmics 사용 설명서의 교차 계정 공유를 참조하세요.

- 자세한 API 내용은 명령 참조GetShare의 섹션을 참조하세요. AWS CLI

### get-variant-import-job

다음 코드 예시에서는 get-variant-import-job을 사용하는 방법을 보여 줍니다.

AWS CLI

변형 가져오기 작업을 보려면

다음 get-variant-import-job 예제에서는 변형 가져오기 작업에 대한 세부 정보를 가져옵니다.

```
aws omics get-variant-import-job \
    --job-id edd7b8ce-xmpl-47e2-bc99-258cac95a508
```

출력:

```
{
```

```
    "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-
W8O1XMPL7QZ",
    "runLeftNormalization": false,
    "status": "IN_PROGRESS",
    "updateTime": "2022-11-23T22:43:05.898309Z"
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 분석을 참조하세요.

- 자세한 API 내용은 명령 참조GetVariantImportJob의 섹션을 참조하세요. AWS CLI

### get-variant-store

다음 코드 예시에서는 get-variant-store을 사용하는 방법을 보여 줍니다.

AWS CLI

변형 스토어를 보려면

다음 get-variant-store 예제에서는 변형 스토어에 대한 세부 정보를 가져옵니다.

```
aws omics get-variant-store \
    --name my_var_store
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 분석을 참조하세요.

- 자세한 API 내용은 명령 참조GetVariantStore의 섹션을 참조하세요. AWS CLI

### get-workflow

다음 코드 예시에서는 get-workflow을 사용하는 방법을 보여 줍니다.

AWS CLI

워크플로를 보려면

다음 get-workflow 예제에서는 ID가 인 워크플로에 대한 세부 정보를 가져옵니다1234567.

```
aws omics get-workflow \
    --id 1234567
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 워크플로를 참조하세요.

- 자세한 API 내용은 명령 참조GetWorkflow의 섹션을 참조하세요. AWS CLI

### `list-annotation-import-jobs`

다음 코드 예시에서는 `list-annotation-import-jobs`을 사용하는 방법을 보여 줍니다.

AWS CLI

주석 가져오기 작업 목록을 가져오려면

다음은 주석 가져오기 작업 목록을 `list-annotation-import-jobs` 가져옵니다.

```
aws omics list-annotation-import-jobs
```

출력:

```
{
    "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-W8O1XMPL7QZ",
            "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-W8O1XMPL7QZ",
            "runLeftNormalization": false,
            "status": "FAILED",
            "updateTime": "2022-11-30T00:47:01.706325Z"
        }
    ]
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics Analytics를 참조하세요.

- 자세한 API 내용은 명령 참조ListAnnotationImportJobs의 섹션을 참조하세요. AWS CLI

### `list-annotation-store-versions`

다음 코드 예시에서는 `list-annotation-store-versions`을 사용하는 방법을 보여 줍니다.

AWS CLI

주석 저장소의 모든 버전을 나열합니다.

다음 `list-annotation-store-versions` 예제에서는 주석 저장소에 있는 모든 버전을 나열합니다.

```
aws omics list-annotation-store-versions \
    --name my_annotation_store
```

출력:

```
{
    "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",
        "creation Time": "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
     }

}
```

자세한 내용은 AWS HealthOmics 사용 설명서의 새 버전의 주석 스토어 생성을 참조하세요.

- 자세한 API 내용은 명령 참조ListAnnotationStoreVersions의 섹션을 참조하세요. AWS CLI

### `list-annotation-stores`

다음 코드 예시에서는 `list-annotation-stores`을 사용하는 방법을 보여 줍니다.

AWS CLI

주석 저장소 목록을 가져오려면

다음 `list-annotation-stores` 예제에서는 주석 저장소 목록을 가져옵니다.

```
aws omics list-annotation-stores
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics Analytics를 참조하세요.

- 자세한 API 내용은 명령 참조ListAnnotationStores의 섹션을 참조하세요. AWS CLI

### `list-multipart-read-set-uploads`

다음 코드 예시에서는 `list-multipart-read-set-uploads`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 멀티파트 읽기 세트 업로드 및 해당 상태를 나열합니다.

다음 `list-multipart-read-set-uploads` 예제에서는 모든 멀티파트 읽기 세트 업로드와 해당 상태를 나열합니다.

```
aws omics list-multipart-read-set-uploads \
    --sequence-store-id 0123456789
```

출력:

```
{
```

```
"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"
        }
    ]
}
```


자세한 내용은 AWS HealthOmics 사용 설명서의 시퀀스 스토어에 직접 업로드를 참조하세요.

- 자세한 API 내용은 명령 참조ListMultipartReadSetUploads의 섹션을 참조하세요. AWS CLI

### `list-read-set-activation-jobs`

다음 코드 예시에서는 `list-read-set-activation-jobs`을 사용하는 방법을 보여 줍니다.

AWS CLI

읽기 세트 활성화 작업 목록을 가져오려면

다음 `list-read-set-activation-jobs` 예제에서는 ID가 인 시퀀스 스토어의 활성화 작업 목록을 가져옵니다1234567890.

```
aws omics list-read-set-activation-jobs \
    --sequence-store-id 1234567890
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 스토리지를 참조하세요.

- 자세한 API 내용은 명령 참조ListReadSetActivationJobs의 섹션을 참조하세요. AWS CLI

### list-read-set-export-jobs

다음 코드 예시에서는 list-read-set-export-jobs을 사용하는 방법을 보여 줍니다.

AWS CLI

읽기 세트 내보내기 작업 목록을 가져오려면

다음 list-read-set-export-jobs 예제에서는 ID가 인 시퀀스 스토어의 내보내기 작업 목록을 가져옵니다1234567890.

```
aws omics list-read-set-export-jobs \
    --sequence-store-id 1234567890
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 스토리지를 참조하세요.

- 자세한 API 내용은 명령 참조ListReadSetExportJobs의 섹션을 참조하세요. AWS CLI

### list-read-set-import-jobs

다음 코드 예시에서는 list-read-set-import-jobs을 사용하는 방법을 보여 줍니다.

AWS CLI

읽기 세트 가져오기 작업 목록을 가져오려면

다음 `list-read-set-import-jobs` 예제에서는 ID가 인 시퀀스 스토어의 가져오기 작업 목록을 가져옵니다1234567890.

```
aws omics list-read-set-import-jobs \
    --sequence-store-id 1234567890
```

출력:

```
{
    "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-W8O1XMPL7QZ",
            "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-W8O1XMPL7QZ",
            "sequenceStoreId": "1234567890",
            "status": "COMPLETED_WITH_FAILURES"
        }
    ]
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 스토리지를 참조하세요.

- 자세한 API 내용은 명령 참조ListReadSetImportJobs의 섹션을 참조하세요. AWS CLI

### `list-read-set-upload-parts`

다음 코드 예시에서는 `list-read-set-upload-parts`을 사용하는 방법을 보여 줍니다.

AWS CLI

시퀀스 스토어에 대해 요청된 멀티파트 업로드의 모든 부분을 나열합니다.

다음 `list-read-set-upload-parts` 예제에서는 시퀀스 스토어에 대해 요청된 멀티파트 업로드의 모든 부분을 나열합니다.

```
aws omics list-read-set-upload-parts \
    --sequence-store-id 0123456789 \
    --upload-id 1122334455 \
    --part-source SOURCE1
```

출력:

```
{
    "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"
        }
      ]

}
```

자세한 내용은 AWS HealthOmics 사용 설명서의 시퀀스 스토어에 직접 업로드를 참조하세요.

- 자세한 API 내용은 명령 참조ListReadSetUploadParts의 섹션을 참조하세요. AWS CLI

### `list-read-sets`

다음 코드 예시에서는 `list-read-sets`을 사용하는 방법을 보여 줍니다.

AWS CLI

읽기 세트 목록을 가져오려면

다음 `list-read-sets` 예제에서는 ID가 인 시퀀스 스토어의 읽기 세트 목록을 가져옵니다1234567890.

```
aws omics list-read-sets \
    --sequence-store-id 1234567890
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 스토리지를 참조하세요.

- 자세한 API 내용은 명령 참조ListReadSets의 섹션을 참조하세요. AWS CLI

### `list-reference-import-jobs`

다음 코드 예시에서는 `list-reference-import-jobs`을 사용하는 방법을 보여 줍니다.

AWS CLI

참조 가져오기 작업 목록을 가져오려면

다음 list-reference-import-jobs 예제에서는 ID가 인 참조 스토어에 대한 참조 가져오기 작업 목록을 가져옵니다1234567890.

```
aws omics list-reference-import-jobs \
    --reference-store-id 1234567890
```

출력:

```
{
    "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-W8O1XMPL7QZ",
            "status": "COMPLETED"
        },
        {
            "creationTime": "2022-11-23T20:34:03.250Z",
            "id": "1234567890",
            "referenceStoreId": "1234567890",
            "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-
serviceRole-W8O1XMPL7QZ",
            "status": "IN_PROGRESS"
        }
    ]
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 스토리지를 참조하세요.

- 자세한 API 내용은 명령 참조ListReferenceImportJobs의 섹션을 참조하세요. AWS CLI

### list-reference-stores

다음 코드 예시에서는 list-reference-stores을 사용하는 방법을 보여 줍니다.

AWS CLI

참조 스토어 목록을 가져오려면

다음 list-reference-stores 예제에서는 참조 스토어 목록을 가져옵니다.

```
aws omics list-reference-stores
```

출력:

```
{
    "referenceStores": [
        {
            "arn": "arn:aws:omics:us-west-2:123456789012:referenceStore/1234567890",
            "creationTime": "2022-11-22T22:13:25.947Z",
            "id": "1234567890",
            "name": "my-ref-store"
        }
    ]
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 스토리지를 참조하세요.

- 자세한 API 내용은 명령 참조ListReferenceStores의 섹션을 참조하세요. AWS CLI

### `list-references`

다음 코드 예시에서는 `list-references`을 사용하는 방법을 보여 줍니다.

AWS CLI

참조 목록을 가져오려면

다음 `list-references` 예제에서는 ID가 인 참조 스토어의 유전체 참조 목록을 가져옵니다`1234567890`.

```
aws omics list-references \
    --reference-store-id 1234567890
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 스토리지를 참조하세요.

- 자세한 API 내용은 명령 참조ListReferences의 섹션을 참조하세요. AWS CLI

### `list-run-groups`

다음 코드 예시에서는 `list-run-groups`을 사용하는 방법을 보여 줍니다.

AWS CLI

실행 그룹 목록을 가져오려면

다음 `list-run-groups` 예제에서는 실행 그룹 목록을 가져옵니다.

```
aws omics list-run-groups
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 워크플로를 참조하세요.

- 자세한 API 내용은 명령 참조ListRunGroups의 섹션을 참조하세요. AWS CLI

### list-run-tasks

다음 코드 예시에서는 list-run-tasks을 사용하는 방법을 보여 줍니다.

AWS CLI

작업 목록을 가져오려면

다음 list-run-tasks 예제에서는 워크플로 실행에 대한 작업 목록을 가져옵니다.

```
aws omics list-run-tasks \
    --id 1234567
```

출력:

```
{
    "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"
        }
    ]
```

```
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 워크플로를 참조하세요.

- 자세한 API 내용은 명령 참조ListRunTasks의 섹션을 참조하세요. AWS CLI

### `list-runs`

다음 코드 예시에서는 `list-runs`을 사용하는 방법을 보여 줍니다.

AWS CLI

워크플로 실행 목록을 가져오려면

다음 `list-runs` 예제에서는 워크플로 실행 목록을 가져옵니다.

```
aws omics list-runs
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 워크플로를 참조하세요.

- 자세한 API 내용은 명령 참조ListRuns의 섹션을 참조하세요. AWS CLI

### `list-sequence-stores`

다음 코드 예시에서는 `list-sequence-stores`을 사용하는 방법을 보여 줍니다.

AWS CLI

시퀀스 스토어 목록을 가져오려면

다음 `list-sequence-stores` 예제에서는 시퀀스 스토어 목록을 가져옵니다.

```
aws omics list-sequence-stores
```

출력:

```
{
    "sequenceStores": [
        {
            "arn": "arn:aws:omics:us-west-2:123456789012:sequenceStore/1234567890",
            "creationTime": "2022-11-23T01:24:33.629Z",
            "id": "1234567890",
            "name": "my-seq-store"
        }
    ]
```

```
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 스토리지를 참조하세요.

- 자세한 API 내용은 명령 참조ListSequenceStores의 섹션을 참조하세요. AWS CLI

### `list-shares`

다음 코드 예시에서는 `list-shares`을 사용하는 방법을 보여 줍니다.

AWS CLI

HealthOmics 분석 데이터의 사용 가능한 공유를 나열하려면

다음 `list-shares` 예제에서는 리소스 소유자에 대해 생성된 모든 공유를 나열합니다.

```
aws omics list-shares \
    --resource-owner SELF
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS HealthOmics 사용 설명서의 교차 계정 공유를 참조하세요.

- 자세한 API 내용은 명령 참조ListShares의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

태그 목록을 가져오려면

다음 `list-tags-for-resource` 예제에서는 ID가 인 워크플로의 태그 목록을 가져옵니다1234567.

```
aws omics list-tags-for-resource \
    --resource-arn arn:aws:omics:us-west-2:123456789012:workflow/1234567
```

출력:

```
{
    "tags": {
        "department": "analytics"
    }
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Amazon Omics에서 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### list-variant-import-jobs

다음 코드 예시에서는 list-variant-import-jobs을 사용하는 방법을 보여 줍니다.

AWS CLI

변형 가져오기 작업 목록을 가져오려면

다음 list-variant-import-jobs 예제에서는 변형 가져오기 작업 목록을 가져옵니다.

```
aws omics list-variant-import-jobs
```

출력:

```
{
    "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-W8O1XMPL7QZ",
            "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-W8O1XMPL7QZ",
            "runLeftNormalization": false,
            "status": "COMPLETED",
            "updateTime": "2022-11-23T22:45:26.009880Z"
        }
    ]
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 분석을 참조하세요.

- 자세한 API 내용은 명령 참조ListVariantImportJobs의 섹션을 참조하세요. AWS CLI

### list-variant-stores

다음 코드 예시에서는 list-variant-stores을 사용하는 방법을 보여 줍니다.

AWS CLI

변형 스토어 목록을 가져오려면

다음 list-variant-stores 예제에서는 변형 스토어 목록을 가져옵니다.

```
aws omics list-variant-stores
```

출력:

```
{
    "variantStores": [
        {
            "creationTime": "2022-11-23T22:09:07.534499Z",
            "id": "02dexmplcfdd",
            "name": "my_var_store",
            "reference": {
                "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890"
            },
            "status": "CREATING",
            "storeArn": "arn:aws:omics:us-west-2:123456789012:variantStore/
my_var_store",
            "storeSizeBytes": 0,
            "updateTime": "2022-11-23T22:09:24.931711Z"
        },
        {
            "creationTime": "2022-09-23T23:00:09.140265Z",
            "id": "8777xmpl1a24",
            "name": "myvstore0",
            "status": "ACTIVE",
            "storeArn": "arn:aws:omics:us-west-2:123456789012:variantStore/
myvstore0",
            "storeSizeBytes": 0,
            "updateTime": "2022-09-23T23:03:26.013220Z"
        }
    ]
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics Analytics를 참조하세요.

- 자세한 API 내용은 명령 참조ListVariantStores의 섹션을 참조하세요. AWS CLI

### `list-workflows`

다음 코드 예시에서는 `list-workflows`을 사용하는 방법을 보여 줍니다.

AWS CLI

워크플로 목록을 가져오려면

다음 `list-workflows` 예제에서는 워크플로 목록을 가져옵니다.

```
aws omics list-workflows
```

출력:

```
{
    "items": [
        {
            "arn": "arn:aws:omics:us-west-2:123456789012:workflow/1234567",
            "creationTime": "2022-09-23T23:08:22.041227Z",
            "digest": "nSCNo/qMWFxmplXpUdokXJnwgneOaxyyc2YOxVxrJTE=",
            "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"
        }
    ]
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 워크플로를 참조하세요.

- 자세한 API 내용은 명령 참조ListWorkflows의 섹션을 참조하세요. AWS CLI

### start-annotation-import-job

다음 코드 예시에서는 start-annotation-import-job을 사용하는 방법을 보여 줍니다.

AWS CLI

주석을 가져오려면

다음 start-annotation-import-job 예제에서는 Amazon S3에서 주석을 가져옵니다.

```
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
```

출력:

```
{
    "jobId": "984162c7-xmpl-4d23-ab47-286f7950bfbf"
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics Analytics를 참조하세요.

- 자세한 API 내용은 명령 참조StartAnnotationImportJob의 섹션을 참조하세요. AWS CLI

### start-read-set-activation-job

다음 코드 예시에서는 start-read-set-activation-job을 사용하는 방법을 보여 줍니다.

AWS CLI

아카이브된 읽기 세트를 활성화하려면

다음 start-read-set-activation-job 예제에서는 두 개의 읽기 세트를 활성화합니다.

```
aws omics start-read-set-activation-job \
    --sequence-store-id 1234567890 \
    --sources readSetId=1234567890 readSetId=1234567890
```

출력:

```
{
    "creationTime": "2022-12-06T22:35:10.100Z",
    "id": "1234567890",
    "sequenceStoreId": "1234567890",
    "status": "SUBMITTED"
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 스토리지를 참조하세요.

- 자세한 API 내용은 명령 참조StartReadSetActivationJob의 섹션을 참조하세요. AWS CLI

### `start-read-set-export-job`

다음 코드 예시에서는 `start-read-set-export-job`을 사용하는 방법을 보여 줍니다.

AWS CLI

읽기 세트를 내보내려면

다음 `start-read-set-export-job` 예제에서는 두 개의 읽기 세트를 Amazon S3로 내보냅니다.

```
   aws omics start-read-set-export-job \
       --sequence-store-id 1234567890 \
       --sources readSetId=1234567890 readSetId=1234567890 \
       --role-arn arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W8O1XMPL7QZ
\
       --destination s3://omics-artifacts-01d6xmpl4e72dd32/read-set-export/
```

출력:

```
{
    "creationTime": "2022-12-06T22:37:18.612Z",
    "destination": "s3://omics-artifacts-01d6xmpl4e72dd32/read-set-export/",
    "id": "1234567890",
    "sequenceStoreId": "1234567890",
    "status": "SUBMITTED"
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 스토리지를 참조하세요.

- 자세한 API 내용은 명령 참조StartReadSetExportJob의 섹션을 참조하세요. AWS CLI

### start-read-set-import-job

다음 코드 예시에서는 start-read-set-import-job을 사용하는 방법을 보여 줍니다.

AWS CLI

읽기 세트를 가져오려면

다음 start-read-set-import-job 예제에서는 읽기 세트를 가져옵니다.

```
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.json은 다음 내용을 포함하는 JSON 문서입니다.

```
[
    {
        "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"
    }
]
```

출력:

```
{
    "creationTime": "2022-11-23T01:36:38.158Z",
```

```
    "id": "1234567890",
    "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W801XMPL7QZ",
    "sequenceStoreId": "1234567890",
    "status": "SUBMITTED"
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 스토리지를 참조하세요.

- 자세한 API 내용은 명령 참조StartReadSetImportJob의 섹션을 참조하세요. AWS CLI

### `start-reference-import-job`

다음 코드 예시에서는 `start-reference-import-job`을 사용하는 방법을 보여 줍니다.

AWS CLI

참조 유전체를 가져오려면

다음 `start-reference-import-job` 예제에서는 Amazon S3에서 참조 유전체를 가져옵니다.

```
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
```

출력:

```
{
    "creationTime": "2022-11-22T22:25:41.124Z",
    "id": "1234567890",
    "referenceStoreId": "1234567890",
    "roleArn": "arn:aws:iam::123456789012:role/omics-service-role-serviceRole-
W801XMPL7QZ",
    "status": "SUBMITTED"
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 스토리지를 참조하세요.

- 자세한 API 내용은 명령 참조StartReferenceImportJob의 섹션을 참조하세요. AWS CLI

**start-run**

다음 코드 예시에서는 start-run을 사용하는 방법을 보여 줍니다.

AWS CLI

워크플로를 실행하려면

다음 start-run 예제에서는 ID 를 사용하여 워크플로를 실행합니다1234567.

```
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.json은 다음 내용을 포함하는 JSON 문서입니다.

```
{
    "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"
}
```

출력:

```
{
    "arn": "arn:aws:omics:us-west-2:123456789012:run/1234567",
    "id": "1234567",
    "status": "PENDING",
    "tags": {}
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 워크플로를 참조하세요.

Amazon Omics에서 소스 파일을 로드하려면

서비스별 를 사용하여 Amazon Omics 스토리지에서 소스 파일을 로드할 수도 있습니다URIs. 다음 예제 workflow-inputs.json 파일은 읽기 세트 및 참조 유전체 소스URIs에 Amazon Omics를 사용합니다.

```
{
    "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"
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 워크플로를 참조하세요.

- 자세한 API 내용은 명령 참조StartRun의 섹션을 참조하세요. AWS CLI

### start-variant-import-job

다음 코드 예시에서는 start-variant-import-job을 사용하는 방법을 보여 줍니다.

AWS CLI

변형 파일을 가져오려면

다음 start-variant-import-job 예제에서는 VCF 형식 변형 파일을 가져옵니다.

```
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
```

출력:

```
{
    "jobId": "edd7b8ce-xmpl-47e2-bc99-258cac95a508"
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics Analytics를 참조하세요.

- 자세한 API 내용은 명령 참조StartVariantImportJob의 섹션을 참조하세요. AWS CLI

### tag-resource

다음 코드 예시에서는 tag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 태그를 지정하려면

다음 tag-resource 예제에서는 ID가 인 워크플로에 department 태그를 추가합니다1234567.

```
aws omics tag-resource \
    --resource-arn arn:aws:omics:us-west-2:123456789012:workflow/1234567 \
    --tags department=analytics
```

자세한 내용은 Amazon Omics 개발자 안내서의 Amazon Omics에서 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### untag-resource

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에서 태그를 제거하려면

다음 untag-resource 예제에서는 워크플로에서 department 태그를 제거합니다.

```
aws omics untag-resource \
    --resource-arn arn:aws:omics:us-west-2:123456789012:workflow/1234567 \
    --tag-keys department
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 스토리지를 참조하세요.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-annotation-store

다음 코드 예시에서는 update-annotation-store을 사용하는 방법을 보여 줍니다.

AWS CLI

주석 저장소를 업데이트하려면

다음 update-annotation-store 예제에서는 라는 주석 저장소에 대한 설명을 업데이트합니다my_vcf_store.

```
aws omics update-annotation-store \
    --name my_vcf_store \
    --description "VCF annotation store"
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics Analytics를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateAnnotationStore의 섹션을 참조하세요. AWS CLI

### update-run-group

다음 코드 예시에서는 update-run-group을 사용하는 방법을 보여 줍니다.

AWS CLI

실행 그룹을 업데이트하려면

다음 update-run-group 예제에서는 ID가 인 실행 그룹의 설정을 업데이트합니다1234567.

```
aws omics update-run-group \
    --id 1234567 \
    --max-cpus 10
```

출력:

```
{
    "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": {}
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 워크플로를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateRunGroup의 섹션을 참조하세요. AWS CLI

### update-variant-store

다음 코드 예시에서는 update-variant-store을 사용하는 방법을 보여 줍니다.

AWS CLI

변형 스토어를 업데이트하려면

다음 update-variant-store 예제에서는 라는 변형 스토어에 대한 설명을 업데이트합니다my_var_store.

```
aws omics update-variant-store \
    --name my_var_store \
    --description "variant store"
```

출력:

```
{
    "creationTime": "2022-11-23T22:09:07.534499Z",
    "description": "variant store",
    "id": "02dexmplcfdd",
    "name": "my_var_store",
    "reference": {
        "referenceArn": "arn:aws:omics:us-
west-2:123456789012:referenceStore/1234567890/reference/1234567890"
    },
    "status": "ACTIVE",
    "updateTime": "2022-12-05T18:23:37.686402Z"
}
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 분석을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateVariantStore의 섹션을 참조하세요. AWS CLI

### update-workflow

다음 코드 예시에서는 update-workflow을 사용하는 방법을 보여 줍니다.

AWS CLI

워크플로를 업데이트하려면

다음 update-workflow 예제에서는 ID가 인 워크플로에 대한 설명을 업데이트합니다1234567.

```
aws omics update-workflow \
    --id 1234567 \
    --description "copy workflow"
```

자세한 내용은 Amazon Omics 개발자 안내서의 Omics 스토리지를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateWorkflow의 섹션을 참조하세요. AWS CLI

### upload-read-set-part

다음 코드 예시에서는 upload-read-set-part을 사용하는 방법을 보여 줍니다.

AWS CLI

읽기 세트 부분을 업로드하는 방법.

다음 upload-read-set-part 예제에서는 읽기 세트의 지정된 부분을 업로드합니다.

```
aws omics upload-read-set-part \
    --sequence-store-id 0123456789 \
    --upload-id 1122334455 \
    --part-source SOURCE1 \
    --part-number 1 \
    --payload /path/to/file/read_1_part_1.fastq.gz
```

출력:

```
{
    "checksum": "984979b9928ae8d8622286c4a9cd8e99d964a22d59ed0f5722e1733eb280e635"
}
```

자세한 내용은 AWS HealthOmics 사용 설명서의 시퀀스 스토어에 직접 업로드를 참조하세요.

- 자세한 API 내용은 명령 참조UploadReadSetPart의 섹션을 참조하세요. AWS CLI

## IAM 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다IAM.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### add-client-id-to-open-id-connect-provider

다음 코드 예시에서는 add-client-id-to-open-id-connect-provider을 사용하는 방법을 보여 줍니다.

AWS CLI

Open-ID Connect() 공급자에 클라이언트 ID(대상OIDC)를 추가하려면

다음 `add-client-id-to-open-id-connect-provider` 명령은 라는 OIDC 공급자에 클라이언트 ID`my-application-ID`를 추가합니다`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
```

이 명령은 출력을 생성하지 않습니다.

OIDC 공급자를 생성하려면 `create-open-id-connect-provider` 명령을 사용합니다.

자세한 내용은 AWS IAM 사용 설명서의 OpenID Connect(OIDC) 자격 증명 공급자 생성을 참조하세요.

- 자세한 API 내용은 명령 참조AddClientIdToOpenIdConnectProvider의 섹션을 참조하세요. AWS CLI

### `add-role-to-instance-profile`

다음 코드 예시에서는 `add-role-to-instance-profile`을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스 프로파일에 역할 추가

다음 `add-role-to-instance-profile` 명령은 이름이 `Webserver`인 인스턴스 프로파일에 이름이 `S3Access`인 역할을 추가합니다.

```
aws iam add-role-to-instance-profile \
    --role-name S3Access \
    --instance-profile-name Webserver
```

이 명령은 출력을 생성하지 않습니다.

인스턴스 프로파일을 생성하려면 `create-instance-profile` 명령을 사용합니다.

자세한 내용은 AWS IAM 사용 설명서EC2의 IAM 역할 사용을 참조하세요.

- 자세한 API 내용은 명령 참조AddRoleToInstanceProfile의 섹션을 참조하세요. AWS CLI

### add-user-to-group

다음 코드 예시에서는 add-user-to-group을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 그룹에 사용자를 추가하려면

다음 add-user-to-group 명령은 라는 이름의 IAM 사용자를 라는 이름Bob의 IAM 그룹에 추가합니다Admins.

```
aws iam add-user-to-group \
    --user-name Bob \
    --group-name Admins
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서 의 IAM 사용자 그룹에서 사용자 추가 및 제거를 참조하세요.

- 자세한 API 내용은 명령 참조AddUserToGroup의 섹션을 참조하세요. AWS CLI

### attach-group-policy

다음 코드 예시에서는 attach-group-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 그룹에 관리형 정책을 연결하려면

다음 attach-group-policy 명령은 라는 AWS 관리형 정책을 라는 ReadOnlyAccess IAM 그룹에 연결합니다Finance.

```
aws iam attach-group-policy \
    --policy-arn arn:aws:iam::aws:policy/ReadOnlyAccess \
    --group-name Finance
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 관리형 정책 및 인라인 정책을 참조하세요.

- 자세한 API 내용은 명령 참조AttachGroupPolicy의 섹션을 참조하세요. AWS CLI

### attach-role-policy

다음 코드 예시에서는 attach-role-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

관리형 정책을 IAM 역할에 연결하려면

다음 attach-role-policy 명령은 라는 AWS 관리형 정책을 라는 ReadOnlyAccess IAM 역할에 연결합니다ReadOnlyRole.

```
aws iam attach-role-policy \
    --policy-arn arn:aws:iam::aws:policy/ReadOnlyAccess \
    --role-name ReadOnlyRole
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 관리형 정책 및 인라인 정책을 참조하세요.

- 자세한 API 내용은 명령 참조AttachRolePolicy의 섹션을 참조하세요. AWS CLI

### attach-user-policy

다음 코드 예시에서는 attach-user-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 사용자에게 관리형 정책을 연결하려면

다음 attach-user-policy 명령은 라는 AWS 관리형 정책을 라는 AdministratorAccess IAM 사용자에게 연결합니다Alice.

```
aws iam attach-user-policy \
    --policy-arn arn:aws:iam::aws:policy/AdministratorAccess \
    --user-name Alice
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 관리형 정책 및 인라인 정책을 참조하세요.

- 자세한 API 내용은 명령 참조AttachUserPolicy의 섹션을 참조하세요. AWS CLI

### change-password

다음 코드 예시에서는 change-password을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 사용자의 암호를 변경하려면

IAM 사용자의 암호를 변경하려면 --cli-input-json 파라미터를 사용하여 이전 암호와 새 암호가 포함된 JSON 파일을 전달하는 것이 좋습니다. 이 방법을 사용하면 영숫자가 아닌 문자가 포함된 강력한 암호를 사용할 수 있습니다. 명령줄 파라미터로 전달할 때 영숫자가 아닌 문자가 포함된 암호는 사용하기 어려울 수 있습니다. --cli-input-json 파라미터를 사용하려면 다음 예와 같이 --generate-cli-skeleton 파라미터와 함께 change-password 명령을 사용하는 것으로 시작합니다.

```
aws iam change-password \
    --generate-cli-skeleton > change-password.json
```

이전 명령은 이전 암호와 새 암호를 채우는 데 사용할 수 있는 change-password.json이라는 JSON 파일을 생성합니다. 예를 들어 파일은 다음과 같을 수 있습니다.

```
{
    "OldPassword": "3s0K_;xh4~8XXI",
    "NewPassword": "]35d/{pB9Fo9wJ"
}
```

그런 다음 암호를 변경하려면 change-password 명령을 다시 사용합니다. 이번에는 --cli-input-json 파라미터를 전달하여 JSON 파일을 지정합니다. 다음 change-password 명령은 change-password.json이라는 JSON 파일과 함께 --cli-input-json 파라미터를 사용합니다.

```
aws iam change-password \
    --cli-input-json file://change-password.json
```

이 명령은 출력을 생성하지 않습니다.

이 명령은 IAM 사용자만 호출할 수 있습니다. AWS 계정(루트) 자격 증명을 사용하여 이 명령을 호출하면 명령이 InvalidUserType 오류를 반환합니다.

자세한 내용은 AWS IAM 사용 설명서 에서 IAM 사용자가 자신의 암호를 변경하는 방법을 참조하세요.

- 자세한 API 내용은 명령 참조ChangePassword의 섹션을 참조하세요. AWS CLI

### `create-access-key`

다음 코드 예시에서는 `create-access-key`을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 사용자의 액세스 키를 생성하려면

다음 `create-access-key` 명령은 이름이 인 IAM 사용자의 액세스 키(액세스 키 ID 및 보안 액세스 키)를 생성합니다Bob.

```
aws iam create-access-key \
    --user-name Bob
```

출력:

```
{
    "AccessKey": {
        "UserName": "Bob",
        "Status": "Active",
        "CreateDate": "2015-03-09T18:39:23.411Z",
        "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYzEXAMPLEKEY",
        "AccessKeyId": "AKIAIOSFODNN7EXAMPLE"
    }
}
```

비밀 액세스 키를 안전한 위치에 저장합니다. 손실된 경우 복구할 수 없으며, 새로운 액세스 키를 생성해야 합니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 사용자에 대한 액세스 키 관리를 참조하세요.

- 자세한 API 내용은 명령 참조CreateAccessKey의 섹션을 참조하세요. AWS CLI

### `create-account-alias`

다음 코드 예시에서는 `create-account-alias`을 사용하는 방법을 보여 줍니다.

AWS CLI

계정 별칭 생성

다음 `create-account-alias` 명령은 AWS 계정의 별칭`examplecorp`을 생성합니다.

```
aws iam create-account-alias \
    --account-alias examplecorp
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 AWS 계정 ID 및 해당 별칭을 참조하세요.

- 자세한 API 내용은 명령 참조CreateAccountAlias의 섹션을 참조하세요. AWS CLI

### `create-group`

다음 코드 예시에서는 `create-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 그룹을 생성하려면

다음 `create-group` 명령은 라는 IAM 그룹을 생성합니다`Admins`.

```
aws iam create-group \
    --group-name Admins
```

출력:

```
{
    "Group": {
        "Path": "/",
        "CreateDate": "2015-03-09T20:30:24.940Z",
        "GroupId": "AIDGPMS9RO4H3FEXAMPLE",
        "Arn": "arn:aws:iam::123456789012:group/Admins",
        "GroupName": "Admins"
    }
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM 사용자 그룹 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateGroup의 섹션을 참조하세요. AWS CLI

### create-instance-profile

다음 코드 예시에서는 create-instance-profile을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스 프로파일 생성

다음 create-instance-profile 명령은 이름이 Webserver인 인스턴스 프로파일을 생성합니다.

```
aws iam create-instance-profile \
    --instance-profile-name Webserver
```

출력:

```
{
    "InstanceProfile": {
        "InstanceProfileId": "AIPAJMBYC7DLSPEXAMPLE",
        "Roles": [],
        "CreateDate": "2015-03-09T20:33:19.626Z",
        "InstanceProfileName": "Webserver",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:instance-profile/Webserver"
    }
}
```

인스턴스 프로파일에 역할을 추가하려면 add-role-to-instance-profile 명령을 사용합니다.

자세한 내용은 AWS IAM 사용 설명서EC2의 IAM 역할 사용을 참조하세요.

- 자세한 API 내용은 명령 참조CreateInstanceProfile의 섹션을 참조하세요. AWS CLI

### create-login-profile

다음 코드 예시에서는 create-login-profile을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 사용자의 암호를 생성하려면

IAM 사용자의 암호를 생성하려면 `--cli-input-json` 파라미터를 사용하여 암호가 포함된 JSON 파일을 전달하는 것이 좋습니다. 이 방법을 사용하면 영숫자가 아닌 문자가 포함된 강력한 암호를 생성할 수 있습니다. 명령줄 파라미터로 전달할 때 영숫자가 아닌 문자가 포함된 암호는 만들기 어려울 수 있습니다.

`--cli-input-json` 파라미터를 사용하려면 다음 예와 같이 `--generate-cli-skeleton` 파라미터와 함께 `create-login-profile` 명령을 사용하는 것으로 시작합니다.

```
aws iam create-login-profile \
    --generate-cli-skeleton > create-login-profile.json
```

이전 명령은 후속 `create-login-profile` 명령에 대한 정보를 채우는 데 사용할 수 있는 create-login-profile.json이라는 JSON 파일을 생성합니다. 예:

```
{
    "UserName": "Bob",
    "Password": "&1-3a6u:RA0djs",
    "PasswordResetRequired": true
}
```

그런 다음 IAM 사용자의 암호를 생성하려면 `create-login-profile` 명령을 다시 사용합니다. 이번에는 `--cli-input-json` 파라미터를 전달하여 JSON 파일을 지정합니다. 다음 `create-login-profile` 명령은 create-login-profile.json이라는 JSON 파일과 함께 `--cli-input-json` 파라미터를 사용합니다.

```
aws iam create-login-profile \
    --cli-input-json file://create-login-profile.json
```

출력:

```
{
    "LoginProfile": {
        "UserName": "Bob",
        "CreateDate": "2015-03-10T20:55:40.274Z",
        "PasswordResetRequired": true
    }
}
```

새 암호가 계정 암호 정책을 위반하는 경우 명령은 `PasswordPolicyViolation` 오류를 반환합니다.

이미 암호가 있는 사용자의 암호를 변경하려면 update-login-profile을 사용합니다. 계정의 암호 정책을 설정하려면 update-account-password-policy 명령을 사용합니다.

계정 암호 정책에서 허용하는 경우 IAM 사용자는 change-password 명령을 사용하여 자신의 암호를 변경할 수 있습니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 사용자 암호 관리를 참조하세요.

- 자세한 API 내용은 명령 참조CreateLoginProfile의 섹션을 참조하세요. AWS CLI

### create-open-id-connect-provider

다음 코드 예시에서는 create-open-id-connect-provider을 사용하는 방법을 보여 줍니다.

AWS CLI

OpenID Connect(OIDC) 공급자를 생성하려면

OpenID Connect(OIDC) 공급자를 생성하려면 --cli-input-json 파라미터를 사용하여 필요한 파라미터가 포함된 JSON 파일을 전달하는 것이 좋습니다. OIDC 공급자를 생성할 때 공급자URL의 를 전달해야 하며 는 로 시작해야 URL 합니다https://. 일부 명령줄 환경에서는 콜론(:) 및 슬래시(/) 문자가 특별한 의미를 가지므로 를 명령줄 파라미터URL로 전달하는 것이 어려울 수 있습니다. --cli-input-json 파라미터를 사용하면 이 제한을 피할 수 있습니다.

--cli-input-json 파라미터를 사용하려면 다음 예와 같이 --generate-cli-skeleton 파라미터와 함께 create-open-id-connect-provider 명령을 사용하는 것으로 시작합니다.

```
aws iam create-open-id-connect-provider \
    --generate-cli-skeleton > create-open-id-connect-provider.json
```

이전 명령은 create-open-id-connect-provider.json이라는 JSON 파일을 생성하여 후속 create-open-id-connect-provider 명령에 대한 정보를 채우는 데 사용할 수 있습니다. 예:

```
{
    "Url": "https://server.example.com",
    "ClientIDList": [
        "example-application-ID"
    ],
    "ThumbprintList": [
        "c3768084dfb3d2b68b7897bf5f565da8eEXAMPLE"
```

```
    ]
}
```

그런 다음 OpenID Connect(OIDC) 공급자를 생성하려면 `create-open-id-connect-provider` 명령을 다시 사용합니다. 이번에는 `--cli-input-json` 파라미터를 전달하여 JSON 파일을 지정합니다. 다음 `create-open-id-connect-provider` 명령은 create-open-id-connect-provider.json이라는 JSON 파일과 함께 `--cli-input-json` 파라미터를 사용합니다.

```
aws iam create-open-id-connect-provider \
    --cli-input-json file://create-open-id-connect-provider.json
```

출력:

```
{
    "OpenIDConnectProviderArn": "arn:aws:iam::123456789012:oidc-provider/
server.example.com"
}
```

OIDC 공급자에 대한 자세한 내용은 AWS IAM 사용 설명서의 OpenID Connect(OIDC) 자격 증명 공급자 생성을 참조하세요.

OIDC 공급자의 지문을 얻는 방법에 대한 자세한 내용은 AWS IAM 사용 설명서의 OpenID Connect Identity Provider에 대한 지문 얻기를 참조하세요.

- 자세한 API 내용은 명령 참조CreateOpenIdConnectProvider의 섹션을 참조하세요. AWS CLI

### `create-policy-version`

다음 코드 예시에서는 `create-policy-version`을 사용하는 방법을 보여 줍니다.

AWS CLI

관리형 정책의 새 버전 생성

이 예제에서는 ARN`arn:aws:iam::123456789012:policy/MyPolicy`를 기본 v2 버전으로 설정한 새 버전의 IAM 정책을 생성합니다.

```
aws iam create-policy-version \
    --policy-arn arn:aws:iam::123456789012:policy/MyPolicy \
    --policy-document file://NewPolicyVersion.json \
```

```
    --set-as-default
```

출력:

```
{
    "PolicyVersion": {
        "CreateDate": "2015-06-16T18:56:03.721Z",
        "VersionId": "v2",
        "IsDefaultVersion": true
    }
}
```

자세한 내용은 AWS IAM 사용 설명서의 버전 관리 IAM 정책을 참조하세요.

- 자세한 API 내용은 명령 참조CreatePolicyVersion의 섹션을 참조하세요. AWS CLI

### `create-policy`

다음 코드 예시에서는 `create-policy`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 고객 관리형 정책 생성

다음 명령은 이름이 `my-policy`인 고객 관리형 정책을 생성합니다.

```
aws iam create-policy \
    --policy-name my-policy \
    --policy-document file://policy
```

파일은 현재 폴더의 JSON 문서로`policy`, 라는 Amazon S3 버킷의 `shared` 폴더에 대한 읽기 전용 액세스 권한을 부여합니다`my-bucket`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:Get*",
                "s3:List*"
```

```
            ],
            "Resource": [
                "arn:aws:s3:::my-bucket/shared/*"
            ]
        }
    ]
}
```

출력:

```
{
    "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"
    }
}
```

파일을 문자열 파라미터의 입력으로 사용하는 방법에 대한 자세한 내용은 AWS CLI 사용 설명서의 에 대한 파라미터 값 지정 AWS CLI을 참조하세요.

예제 2: 설명이 포함된 고객 관리형 정책 생성

다음 명령은 변경 불가능한 설명이 포함된 이름이 `my-policy`인 고객 관리형 정책을 생성합니다.

```
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"
```

파일은 현재 폴더의 JSON 문서`policy.json`로, 라는 Amazon S3 버킷에 대한 모든 Put, List 및 Get 작업에 대한 액세스 권한을 부여합니다`my-bucket`.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                 "s3:ListBucket*",
                 "s3:PutBucket*",
                 "s3:GetBucket*"
             ],
             "Resource": [
                 "arn:aws:s3:::my-bucket"
             ]
         }
     ]
 }
```

출력:

```
{
    "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"
    }
}
```

Idenity 기반 정책에 대한 자세한 내용은 AWS IAM 사용 설명서의 자격 증명 기반 정책 및 리소스 기반 정책을 참조하세요.

예제 3: 태그가 포함된 고객 관리형 정책 생성

다음 명령은 태그가 포함된 이름이 `my-policy`인 고객 관리형 정책을 생성합니다. 이 예제에서는 형식이 지정된 JSON태그인 와 함께 `--tags` 파라미터 플래그를 사용합니다`'{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location", "Value": "Seattle"}'`. 또는 `--tags` 플래그를 짧은 형식의 태그

('Key=Department,Value=Accounting Key=Location,Value=Seattle')에 사용할 수도 있습니다.

```
aws iam create-policy \
    --policy-name my-policy \
    --policy-document file://policy.json \
    --tags '{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location",
 "Value": "Seattle"}'
```

파일은 현재 폴더의 JSON 문서policy.json로, 라는 Amazon S3 버킷에 대한 모든 Put, List 및 Get 작업에 대한 액세스 권한을 부여합니다my-bucket.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                 "s3:ListBucket*",
                 "s3:PutBucket*",
                 "s3:GetBucket*"
             ],
             "Resource": [
                 "arn:aws:s3:::my-bucket"
             ]
         }
     ]
 }
```

출력:

```
{
    "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"
            {

        ]
    }
}
```

태깅 정책에 대한 자세한 내용은 AWS IAM 사용 설명서의 고객 관리형 정책 태깅을 참조하세요.

- 자세한 API 내용은 명령 참조CreatePolicy의 섹션을 참조하세요. AWS CLI

### `create-role`

다음 코드 예시에서는 `create-role`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: IAM 역할 생성

다음 `create-role` 명령은 이름이 `Test-Role`인 역할을 생성하고 해당 역할에 신뢰 정책을 연결합니다.

```
aws iam create-role \
    --role-name Test-Role \
    --assume-role-policy-document file://Test-Role-Trust-Policy.json
```

출력:

```
{
    "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"
    }
}
```

신뢰 정책은 Test-Role-Trust-Policy.json 파일의 JSON 문서로 정의됩니다. (파일 이름과 확장자는 중요하지 않습니다.) 신뢰 정책에서 보안 주체를 지정해야 합니다.

역할에 권한 정책을 연결하려면 `put-role-policy` 명령을 사용합니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 역할 생성을 참조하세요.

예제 2: 지정된 최대 세션 기간으로 IAM 역할을 생성하려면

다음 `create-role` 명령은 이름이 `Test-Role`인 역할을 생성하고 최대 세션 지속 시간을 7,200초(2시간)로 설정합니다.

```
aws iam create-role \
    --role-name Test-Role \
    --assume-role-policy-document file://Test-Role-Trust-Policy.json \
    --max-session-duration 7200
```

출력:

```
{
    "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"
                }
            ]
```

```
        }
    }
}
```

자세한 내용은 AWS IAM 사용 설명서의 역할 최대 세션 기간 수정(AWS API)을 참조하세요.

예제 3: 태그를 사용하여 IAM 역할 생성

다음 명령은 태그가 Test-Role 있는 IAM 역할을 생성합니다. 이 예제에서는 형식이 지정된 JSON태그인 와 함께 --tags 파라미터 플래그를 사용합니다'{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location", "Value": "Seattle"}'. 또는 --tags 플래그를 짧은 형식의 태그('Key=Department,Value=Accounting Key=Location,Value=Seattle')에 사용할 수도 있습니다.

```
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"}'
```

출력:

```
{
    "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"
            }
        ]
    }
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM 역할 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조CreateRole의 섹션을 참조하세요. AWS CLI

### `create-saml-provider`

다음 코드 예시에서는 `create-saml-provider`을 사용하는 방법을 보여 줍니다.

AWS CLI

SAML 공급자를 생성하려면

이 예제에서는 IAM 이름이 인 에 새 SAML 공급자를 생성합니다`MySAMLProvider`. 파일 에 있는 SAML 메타데이터 문서에 설명되어 있습니다`SAMLMetaData.xml`.

```
aws iam create-saml-provider \
    --saml-metadata-document file://SAMLMetaData.xml \
    --name MySAMLProvider
```

출력:

```
{
    "SAMLProviderArn": "arn:aws:iam::123456789012:saml-provider/MySAMLProvider"
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM SAML 자격 증명 공급자 생성을 참조하세요.

- 자세한 API 내용은 AWS CLI 명령 참조의 CreateSAMLProvider를 참조하세요.

### create-service-linked-role

다음 코드 예시에서는 create-service-linked-role을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스 연결 역할 생성

다음 create-service-linked-role 예제에서는 지정된 서비스에 대한 AWS 서비스 연결 역할을 생성하고 지정된 설명을 연결합니다.

```
aws iam create-service-linked-role \
    --aws-service-name lex.amazonaws.com \
    --description "My service-linked role to support Lex"
```

출력:

```
{
    "Role": {
        "Path": "/aws-service-role/lex.amazonaws.com/",
        "RoleName": "AWSServiceRoleForLexBots",
        "RoleId": "AROA1234567890EXAMPLE",
        "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"
                        ]
                    }
                }
            ]
        }
    }
}
```

자세한 내용은 AWS IAM 사용 설명서의 서비스 연결 역할 사용을 참조하세요.

- 자세한 API 내용은 명령 참조CreateServiceLinkedRole의 섹션을 참조하세요. AWS CLI

### create-service-specific-credential

다음 코드 예시에서는 create-service-specific-credential을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자에 대한 서비스별 자격 증명 세트 생성

다음 create-service-specific-credential 예제에서는 구성된 서비스에만 액세스하는 데 사용할 수 있는 사용자 이름과 암호를 생성합니다.

```
aws iam create-service-specific-credential \
    --user-name sofia \
    --service-name codecommit.amazonaws.com
```

출력:

```
{
    "ServiceSpecificCredential": {
        "CreateDate": "2019-04-18T20:45:36+00:00",
        "ServiceName": "codecommit.amazonaws.com",
        "ServiceUserName": "sofia-at-123456789012",
        "ServicePassword": "k1zPZM6uVxMQ3oxqgoYlNuJPyRTZ1vREs76zTQE3eJk=",
        "ServiceSpecificCredentialId": "ACCAEXAMPLE123EXAMPLE",
        "UserName": "sofia",
        "Status": "Active"
    }
}
```

자세한 내용은 AWS CodeCommit 사용 설명서의 에 대한 HTTPS 연결을 위한 Git 자격 증명 생성을 CodeCommit 참조하세요.

- 자세한 API 내용은 명령 참조CreateServiceSpecificCredential의 섹션을 참조하세요. AWS CLI

### create-user

다음 코드 예시에서는 create-user을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: IAM 사용자 생성

다음 `create-user` 명령은 현재 계정에 Bob 이름이 지정된 IAM 사용자를 생성합니다.

```
aws iam create-user \
    --user-name Bob
```

출력:

```
{
    "User": {
        "UserName": "Bob",
        "Path": "/",
        "CreateDate": "2023-06-08T03:20:41.270Z",
        "UserId": "AIDAIOSFODNN7EXAMPLE",
        "Arn": "arn:aws:iam::123456789012:user/Bob"
    }
}
```

자세한 내용은 AWS IAM 사용 설명서의 AWS 계정에서 IAM 사용자 생성을 참조하세요.

예제 2: 지정된 경로에서 IAM 사용자를 생성하려면

다음 `create-user` 명령은 지정된 경로Bob에 이름이 지정된 IAM 사용자를 생성합니다.

```
aws iam create-user \
    --user-name Bob \
    --path /division_abc/subdivision_xyz/
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM 식별자를 참조하세요.

예제 3: 태그가 있는 IAM 사용자 생성

다음 `create-user` 명령은 태그로 이름이 Bob 지정된 IAM 사용자를 생성합니다. 이 예제에서는 형식이 지정된 JSON태그인 와 함께 `--tags` 파라미터 플래그를 사용합니다`'{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location", "Value": "Seattle"}'`. 또는 `--tags` 플래그를 짧은 형식의 태그 (`'Key=Department,Value=Accounting Key=Location,Value=Seattle'`)에 사용할 수도 있습니다.

```
aws iam create-user \
    --user-name Bob \
    --tags '{"Key": "Department", "Value": "Accounting"}' '{"Key": "Location",
 "Value": "Seattle"}'
```

출력:

```
{
    "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"
            }
        ]
    }
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM 사용자 태그 지정을 참조하세요.

예제 3: 설정된 권한 경계가 있는 IAM 사용자를 생성하려면

다음 create-user 명령은 AmazonS3FullAccess의 권한 경계Bob로 라는 IAM 사용자를 생성합니다.

```
aws iam create-user \
    --user-name Bob \
    --permissions-boundary arn:aws:iam::aws:policy/AmazonS3FullAccess
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM엔터티에 대한 권한 경계를 참조하세요.

- 자세한 API 내용은 명령 참조CreateUser의 섹션을 참조하세요. AWS CLI

### create-virtual-mfa-device

다음 코드 예시에서는 create-virtual-mfa-device을 사용하는 방법을 보여 줍니다.

AWS CLI

가상 MFA 디바이스를 생성하려면

이 예제에서는 라는 새 가상 MFA 디바이스를 생성합니다BobsMFADevice. 부트스트랩 정보가 포함된 QRCode.png라는 파일을 생성하여 C:/ 디렉터리에 배치합니다. 이 예제에서 사용된 부트스트랩 방법은 QRCodePNG입니다.

```
aws iam create-virtual-mfa-device \
    --virtual-mfa-device-name BobsMFADevice \
```

```
    --outfile C:/QRCode.png \
    --bootstrap-method QRCodePNG
```

출력:

```
{
    "VirtualMFADevice": {
        "SerialNumber": "arn:aws:iam::210987654321:mfa/BobsMFADevice"
}
```

자세한 내용은 AWS IAM 사용 설명서의 에서 다중 인증 사용(MFA) AWS을 참조하세요.

- 자세한 API 내용은 명령 참조CreateVirtualMfaDevice의 섹션을 참조하세요. AWS CLI

### deactivate-mfa-device

다음 코드 예시에서는 deactivate-mfa-device을 사용하는 방법을 보여 줍니다.

AWS CLI

MFA 디바이스를 비활성화하려면

이 명령ARNarn:aws:iam::210987654321:mfa/BobsMFADevice은 사용자 와 연결된 를 사용하여 가상 MFA 디바이스를 비활성화합니다Bob.

```
aws iam deactivate-mfa-device \
    --user-name Bob \
    --serial-number arn:aws:iam::210987654321:mfa/BobsMFADevice
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 에서 다중 인증 사용(MFA) AWS을 참조하세요.

- 자세한 API 내용은 명령 참조DeactivateMfaDevice의 섹션을 참조하세요. AWS CLI

### decode-authorization-message

다음 코드 예시에서는 decode-authorization-message을 사용하는 방법을 보여 줍니다.

AWS CLI

권한 부여 실패 메시지를 디코딩하려면

다음 decode-authorization-message 예제에서는 필요한 권한 없이 인스턴스를 시작하려고 할 때 EC2 콘솔에서 반환하는 메시지를 디코딩합니다.

```
aws sts decode-authorization-message \
    --encoded-message lxzA8VEjEvu-s0TTt3PgYCXik9YakOqsrFJGRZR98xNcyWAxwRq14xIvd-
npzbgTevuufCTbjeBAaDARg9cbTK1rJbg3awM33o-Vy3ebPErE2-
mWR9hVYdvX-0zKgVOWF9pWjZaJSMqxB-aLXo-I_8TTvBq88x8IFPbMArNdpu0IjxDjzf22PF3SOE3XvIQ-
_PEO0aUqHCCcsSrFtvxm6yQD1nbm6VTIVrfa0Bzy8lsoMo7SjIaJ2r5vph6SY5vCCwg6o2JKe3hIHTa8zRrDbZSFMkcX
Xx9AYAAIr6bhcis7C__bZh4dlAAWooHFGKgfoJcWGwgdzgbu9hWyVvKTpeot5hsb8qANYjJRCPXTKpi6PZfdijIkwb6g
```

출력은 JSON 텍스트 JSON 프로세서로 구문 분석할 수 있는 텍스트의 단일 줄 문자열로 형식이 지정됩니다.

```
{
    "DecodedMessage": "{\"allowed\":false,\"explicitDeny\":false,\"matchedStatements
\":{\"items\":[]},\"failures\":{\"items\":[]},\"context\":{\"principal
\":{\"id\":\"AIDAV3ZUEFP6J7GY7O6LO\",\"name\":\"chain-user\",\"arn\":
\"arn:aws:iam::403299380220:user/chain-user\"},\"action\":\"ec2:RunInstances\",
\"resource\":\"arn:aws:ec2:us-east-2:403299380220:instance/*\",\"conditions\":
{\"items\":[{\"key\":\"ec2:InstanceMarketType\",\"values\":{\"items\":[{\"value
\":\"on-demand\"}]}},{\"key\":\"aws:Resource\",\"values\":{\"items\":[{\"value
\":\"instance/*\"}]}},{\"key\":\"aws:Account\",\"values\":{\"items\":[{\"value
\":\"403299380220\"}]}},{\"key\":\"ec2:AvailabilityZone\",\"values\":{\"items\":
[{\"value\":\"us-east-2b\"}]}},{\"key\":\"ec2:ebsOptimized\",\"values\":{\"items
\":[{\"value\":\"false\"}]}},{\"key\":\"ec2:IsLaunchTemplateResource\",\"values
\":{\"items\":[{\"value\":\"false\"}]}},{\"key\":\"ec2:InstanceType\",\"values
\":{\"items\":[{\"value\":\"t2.micro\"}]}},{\"key\":\"ec2:RootDeviceType\",
\"values\":{\"items\":[{\"value\":\"ebs\"}]}},{\"key\":\"aws:Region\",\"values
\":{\"items\":[{\"value\":\"us-east-2\"}]}},{\"key\":\"aws:Service\",\"values
\":{\"items\":[{\"value\":\"ec2\"}]}},{\"key\":\"ec2:InstanceID\",\"values\":
{\"items\":[{\"value\":\"*\"}]}},{\"key\":\"aws:Type\",\"values\":{\"items\":
[{\"value\":\"instance\"}]}},{\"key\":\"ec2:Tenancy\",\"values\":{\"items\":
[{\"value\":\"default\"}]}},{\"key\":\"ec2:Region\",\"values\":{\"items\":[{\"value
\":\"us-east-2\"}]}},{\"key\":\"aws:ARN\",\"values\":{\"items\":[{\"value\":
\"arn:aws:ec2:us-east-2:403299380220:instance/*\"}]}}]}}}"
}
```

자세한 내용은 AWS re:Post 에서 EC2 인스턴스 시작 중에 “UnauthorizedOperation” 오류를 수신한 후 권한 부여 실패 메시지를 해독하려면 어떻게 해야 합니까?를 참조하세요.

- 자세한 API 내용은 명령 참조DecodeAuthorizationMessage의 섹션을 참조하세요. AWS CLI

### delete-access-key

다음 코드 예시에서는 delete-access-key을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 사용자의 액세스 키를 삭제하려면

다음 delete-access-key 명령은 이름이 인 IAM 사용자의 지정된 액세스 키(액세스 키 ID 및 보안 액세스 키)를 삭제합니다Bob.

```
aws iam delete-access-key \
    --access-key-id AKIDPMS9RO4H3FEXAMPLE \
    --user-name Bob
```

이 명령은 출력을 생성하지 않습니다.

IAM 사용자에 대해 정의된 액세스 키를 나열하려면 list-access-keys 명령을 사용합니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 사용자 액세스 키 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteAccessKey의 섹션을 참조하세요. AWS CLI

### delete-account-alias

다음 코드 예시에서는 delete-account-alias을 사용하는 방법을 보여 줍니다.

AWS CLI

계정 별칭 삭제

다음 delete-account-alias 명령은 현재 계정의 별칭 mycompany를 제거합니다.

```
aws iam delete-account-alias \
    --account-alias mycompany
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 AWS 계정 ID 및 해당 별칭을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteAccountAlias의 섹션을 참조하세요. AWS CLI

### delete-account-password-policy

다음 코드 예시에서는 delete-account-password-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 계정 암호 정책 삭제

다음 delete-account-password-policy 명령은 현재 계정의 암호 정책을 제거합니다.

```
aws iam delete-account-password-policy
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 사용자에 대한 계정 암호 정책 설정을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteAccountPasswordPolicy의 섹션을 참조하세요. AWS CLI

### delete-group-policy

다음 코드 예시에서는 delete-group-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 그룹에서 정책을 삭제하려면

다음 delete-group-policy 명령은 이름이 Admins인 그룹에서 이름이 ExamplePolicy인 정책을 삭제합니다.

```
aws iam delete-group-policy \
    --group-name Admins \
    --policy-name ExamplePolicy
```

이 명령은 출력을 생성하지 않습니다.

그룹에 연결된 정책을 보려면 list-group-policies 명령을 사용합니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 정책 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteGroupPolicy의 섹션을 참조하세요. AWS CLI

### delete-group

다음 코드 예시에서는 delete-group을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 그룹을 삭제하려면

다음 delete-group 명령은 이름이 인 IAM 그룹을 삭제합니다MyTestGroup.

```
aws iam delete-group \
    --group-name MyTestGroup
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 사용자 그룹 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteGroup의 섹션을 참조하세요. AWS CLI

### delete-instance-profile

다음 코드 예시에서는 delete-instance-profile을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스 프로파일 삭제

다음 delete-instance-profile 명령은 이름이 ExampleInstanceProfile인 인스턴스 프로파일을 삭제합니다.

```
aws iam delete-instance-profile \
    --instance-profile-name ExampleInstanceProfile
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 인스턴스 프로파일 사용을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteInstanceProfile의 섹션을 참조하세요. AWS CLI

### delete-login-profile

다음 코드 예시에서는 delete-login-profile을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 사용자의 암호를 삭제하려면

다음 delete-login-profile 명령은 이름이 인 IAM 사용자의 암호를 삭제합니다Bob.

```
aws iam delete-login-profile \
    --user-name Bob
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 사용자 암호 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteLoginProfile의 섹션을 참조하세요. AWS CLI

### delete-open-id-connect-provider

다음 코드 예시에서는 delete-open-id-connect-provider을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM OpenID Connect 자격 증명 공급자를 삭제하려면

이 예제에서는 IAM OIDC 공급자 에 연결하는 공급자를 삭제합니다example.oidcprovider.com.

```
aws iam delete-open-id-connect-provider \
    --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/
example.oidcprovider.com
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 OpenID Connect(OIDC) 자격 증명 공급자 생성을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteOpenIdConnectProvider의 섹션을 참조하세요. AWS CLI

### delete-policy-version

다음 코드 예시에서는 delete-policy-version을 사용하는 방법을 보여 줍니다.

AWS CLI

관리형 정책의 버전 삭제

이 예제에서는 가 ARN인 정책v2에서 로 식별된 버전을 삭제합니다arn:aws:iam::123456789012:policy/MySamplePolicy.

```
aws iam delete-policy-version \
    --policy-arn arn:aws:iam::123456789012:policy/MyPolicy \
    --version-id v2
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 에서 정책 및 권한을 IAM 참조하세요.

- 자세한 API 내용은 명령 참조DeletePolicyVersion의 섹션을 참조하세요. AWS CLI

### delete-policy

다음 코드 예시에서는 delete-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 정책을 삭제하려면

이 예제ARN에서는 이 인 정책을 삭제합니다arn:aws:iam::123456789012:policy/MySamplePolicy.

```
aws iam delete-policy \
    --policy-arn arn:aws:iam::123456789012:policy/MySamplePolicy
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 에서 정책 및 권한을 IAM 참조하세요.

- 자세한 API 내용은 명령 참조DeletePolicy의 섹션을 참조하세요. AWS CLI

### delete-role-permissions-boundary

다음 코드 예시에서는 delete-role-permissions-boundary을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 역할에서 권한 경계를 삭제하려면

다음 delete-role-permissions-boundary 예제에서는 지정된 IAM 역할에 대한 권한 경계를 삭제합니다. 역할에 권한 경계를 적용하려면 put-role-permissions-boundary 명령을 사용합니다.

```
aws iam delete-role-permissions-boundary \
    --role-name lambda-application-role
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 에서 정책 및 권한을 IAM 참조하세요.

- 자세한 API 내용은 명령 참조DeleteRolePermissionsBoundary의 섹션을 참조하세요. AWS CLI

### delete-role-policy

다음 코드 예시에서는 delete-role-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 역할에서 정책을 제거하려면

다음 delete-role-policy 명령은 이름이 Test-Role인 역할에서 이름이 ExamplePolicy인 정책을 제거합니다.

```
aws iam delete-role-policy \
    --role-name Test-Role \
    --policy-name ExamplePolicy
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 역할 수정을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteRolePolicy의 섹션을 참조하세요. AWS CLI

### delete-role

다음 코드 예시에서는 delete-role을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 역할을 삭제하려면

다음 delete-role 명령은 이름이 Test-Role인 역할을 제거합니다.

```
aws iam delete-role \
    --role-name Test-Role
```

이 명령은 출력을 생성하지 않습니다.

역할을 삭제하려면 먼저 인스턴스 프로파일에서 역할을 제거하고 (remove-role-from-instance-profile), 관리형 정책을 모두 분리하고(detach-role-policy), 역할에 연결된 인라인 정책을 모두 삭제해야 합니다(delete-role-policy).

자세한 내용은 AWS IAM 사용 설명서의 IAM 역할 생성 및 인스턴스 프로파일 사용을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteRole의 섹션을 참조하세요. AWS CLI

### delete-saml-provider

다음 코드 예시에서는 delete-saml-provider을 사용하는 방법을 보여 줍니다.

AWS CLI

SAML 공급자를 삭제하려면

이 예제에서는 가 인 IAM SAML 2.0 공급자ARN를 삭제합니다arn:aws:iam::123456789012:saml-provider/SAMLADFSProvider.

```
aws iam delete-saml-provider \
--saml-provider-arn arn:aws:iam::123456789012:saml-provider/SAMLADFSProvider
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM SAML 자격 증명 공급자 생성을 참조하세요.

- 자세한 API 내용은 AWS CLI 명령 참조의 DeleteSAMLProvider를 참조하세요.

### delete-server-certificate

다음 코드 예시에서는 delete-server-certificate을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 계정에서 서버 인증서를 삭제하려면

다음 delete-server-certificate 명령은 AWS 계정에서 지정된 서버 인증서를 제거합니다.

```
aws iam delete-server-certificate \
    --server-certificate-name myUpdatedServerCertificate
```

이 명령은 출력을 생성하지 않습니다.

AWS 계정에서 사용할 수 있는 서버 인증서를 나열하려면 `list-server-certificates` 명령을 사용합니다.

자세한 내용은 AWS IAM 사용 설명서의 에서 서버 인증서 관리를 IAM 참조하세요.

- 자세한 API 내용은 명령 참조DeleteServerCertificate의 섹션을 참조하세요. AWS CLI

### `delete-service-linked-role`

다음 코드 예시에서는 `delete-service-linked-role`을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스 연결 역할 삭제

다음 `delete-service-linked-role` 예제에서는 더 이상 필요하지 않은 지정된 서비스 연결 역할을 삭제합니다. 삭제는 비동기식으로 이루어집니다. `get-service-linked-role-deletion-status` 명령을 사용하여 삭제 상태를 확인하고 언제 삭제되는지 확인할 수 있습니다.

```
aws iam delete-service-linked-role \
    --role-name AWSServiceRoleForLexBots
```

출력:

```
{
    "DeletionTaskId": "task/aws-service-role/lex.amazonaws.com/
AWSServiceRoleForLexBots/1a2b3c4d-1234-abcd-7890-abcdeEXAMPLE"
}
```

자세한 내용은 AWS IAM 사용 설명서의 서비스 연결 역할 사용을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteServiceLinkedRole의 섹션을 참조하세요. AWS CLI

### `delete-service-specific-credential`

다음 코드 예시에서는 `delete-service-specific-credential`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 요청 사용자의 서비스별 자격 증명 삭제

다음 delete-service-specific-credential 예제에서는 요청을 수행하는 사용자의 지정된 서비스별 자격 증명을 삭제합니다. service-specific-credential-id 는 자격 증명을 생성할 때 제공되며 list-service-specific-credentials 명령을 사용하여 검색할 수 있습니다.

```
aws iam delete-service-specific-credential \
    --service-specific-credential-id ACCAEXAMPLE123EXAMPLE
```

이 명령은 출력을 생성하지 않습니다.

예제 2: 지정된 사용자의 서비스별 자격 증명 삭제

다음 delete-service-specific-credential 예제에서는 지정된 사용자에 대해 지정된 서비스별 자격 증명을 삭제합니다. service-specific-credential-id 는 자격 증명을 생성할 때 제공되며 list-service-specific-credentials 명령을 사용하여 검색할 수 있습니다.

```
aws iam delete-service-specific-credential \
    --user-name sofia \
    --service-specific-credential-id ACCAEXAMPLE123EXAMPLE
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS CodeCommit 사용 설명서의 에 대한 HTTPS 연결을 위한 Git 자격 증명 생성을 CodeCommit 참조하세요.

- 자세한 API 내용은 명령 참조DeleteServiceSpecificCredential의 섹션을 참조하세요. AWS CLI

### delete-signing-certificate

다음 코드 예시에서는 delete-signing-certificate을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 사용자의 서명 인증서를 삭제하려면

다음 delete-signing-certificate 명령은 라는 IAM 사용자에 대해 지정된 서명 인증서를 삭제합니다Bob.

```
aws iam delete-signing-certificate \
    --user-name Bob \
    --certificate-id TA7SMP42TDN5Z26OBPJE7EXAMPLE
```

이 명령은 출력을 생성하지 않습니다.

서명 인증서의 ID를 가져오려면 `list-signing-certificates` 명령을 사용합니다.

자세한 내용은 Amazon EC2 사용 설명서의 서명 인증서 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteSigningCertificate의 섹션을 참조하세요. AWS CLI

### `delete-ssh-public-key`

다음 코드 예시에서는 `delete-ssh-public-key`을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 사용자에게 연결된 SSH 퍼블릭 키를 삭제하려면

다음 `delete-ssh-public-key` 명령은 IAM 사용자 에 연결된 지정된 SSH 퍼블릭 키를 삭제합니다sofia.

```
aws iam delete-ssh-public-key \
    --user-name sofia \
    --ssh-public-key-id APKA123456789EXAMPLE
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 및 SSH와 함께 SSH 키 사용을 CodeCommit 참조하세요.

- 자세한 API 내용은 명령 참조DeleteSshPublicKey의 섹션을 참조하세요. AWS CLI

### `delete-user-permissions-boundary`

다음 코드 예시에서는 `delete-user-permissions-boundary`을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 사용자로부터 권한 경계를 삭제하려면

다음 `delete-user-permissions-boundary` 예제에서는 라는 IAM 사용자에 연결된 권한 경계를 삭제합니다intern. 사용자에게 권한 경계를 적용하려면 `put-user-permissions-boundary` 명령을 사용합니다.

```
aws iam delete-user-permissions-boundary \
```

```
    --user-name intern
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 에서 정책 및 권한을 IAM 참조하세요.

- 자세한 API 내용은 명령 참조DeleteUserPermissionsBoundary의 섹션을 참조하세요. AWS CLI

### delete-user-policy

다음 코드 예시에서는 delete-user-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 사용자로부터 정책을 제거하려면

다음 delete-user-policy 명령은 라는 IAM 사용자로부터 지정된 정책을 제거합니다Bob.

```
aws iam delete-user-policy \
    --user-name Bob \
    --policy-name ExamplePolicy
```

이 명령은 출력을 생성하지 않습니다.

IAM 사용자의 정책 목록을 가져오려면 list-user-policies 명령을 사용합니다.

자세한 내용은 AWS IAM 사용 설명서의 AWS 계정에서 IAM 사용자 생성을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteUserPolicy의 섹션을 참조하세요. AWS CLI

### delete-user

다음 코드 예시에서는 delete-user을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 사용자를 삭제하려면

다음 delete-user 명령은 현재 계정Bob에서 이름이 인 IAM 사용자를 제거합니다.

```
aws iam delete-user \
    --user-name Bob
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 사용자 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteUser의 섹션을 참조하세요. AWS CLI

### delete-virtual-mfa-device

다음 코드 예시에서는 delete-virtual-mfa-device을 사용하는 방법을 보여 줍니다.

AWS CLI

가상 MFA 디바이스를 제거하려면

다음 delete-virtual-mfa-device 명령은 현재 계정에서 지정된 MFA 디바이스를 제거합니다.

```
aws iam delete-virtual-mfa-device \
    --serial-number arn:aws:iam::123456789012:mfa/MFATest
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 MFA 디바이스 비활성화를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteVirtualMfaDevice의 섹션을 참조하세요. AWS CLI

### detach-group-policy

다음 코드 예시에서는 detach-group-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

그룹에서 정책 분리

이 예제에서는 라는 그룹에서 ARNarn:aws:iam::123456789012:policy/TesterAccessPolicy를 사용하여 관리형 정책을 제거합니다Testers.

```
aws iam detach-group-policy \
    --group-name Testers \
    --policy-arn arn:aws:iam::123456789012:policy/TesterAccessPolicy
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 사용자 그룹 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DetachGroupPolicy의 섹션을 참조하세요. AWS CLI

### detach-role-policy

다음 코드 예시에서는 detach-role-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

역할에서 정책 분리

이 예제에서는 라는 역할ARNarn:aws:iam::123456789012:policy/FederatedTesterAccessPolicy에서 를 사용하여 관리형 정책을 제거합니다FedTesterRole.

```
aws iam detach-role-policy \
    --role-name FedTesterRole \
    --policy-arn arn:aws:iam::123456789012:policy/FederatedTesterAccessPolicy
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 역할 수정을 참조하세요.

- 자세한 API 내용은 명령 참조DetachRolePolicy의 섹션을 참조하세요. AWS CLI

### detach-user-policy

다음 코드 예시에서는 detach-user-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자에서 정책 분리

이 예제에서는 사용자 ARN arn:aws:iam::123456789012:policy/TesterPolicy 에서 를 사용하여 관리형 정책을 제거합니다Bob.

```
aws iam detach-user-policy \
    --user-name Bob \
    --policy-arn arn:aws:iam::123456789012:policy/TesterPolicy
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 사용자 권한 변경을 참조하세요.

- 자세한 API 내용은 명령 참조DetachUserPolicy의 섹션을 참조하세요. AWS CLI

### enable-mfa-device

다음 코드 예시에서는 enable-mfa-device을 사용하는 방법을 보여 줍니다.

AWS CLI

MFA 디바이스를 활성화하려면

create-virtual-mfa-device 명령을 사용하여 새 가상 MFA 디바이스를 생성한 후 사용자에게 MFA 디바이스를 할당할 수 있습니다. 다음 enable-mfa-device 예제에서는 일련 번호가 인 MFA 디바이스arn:aws:iam::210987654321:mfa/BobsMFADevice를 사용자에게 할당합니다Bob. 또한 명령은 가상 디바이스에서 처음 두 코드를 순서대로 AWS 포함하여 MFA 디바이스를와 동기화합니다.

```
aws iam enable-mfa-device \
    --user-name Bob \
    --serial-number arn:aws:iam::210987654321:mfa/BobsMFADevice \
    --authentication-code1 123456 \
    --authentication-code2 789012
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 가상 다중 인증(MFA) 디바이스 활성화를 참조하세요.

- 자세한 API 내용은 명령 참조EnableMfaDevice의 섹션을 참조하세요. AWS CLI

### generate-credential-report

다음 코드 예시에서는 generate-credential-report을 사용하는 방법을 보여 줍니다.

AWS CLI

보안 인증 보고서 생성

다음 예제에서는 AWS 계정에 대한 자격 증명 보고서를 생성하려고 시도합니다.

```
aws iam generate-credential-report
```

출력:

```
{
    "State":  "STARTED",
    "Description": "No report exists. Starting a new report generation task"
}
```

자세한 내용은 AWS IAM 사용 설명서의 AWS 계정에 대한 자격 증명 보고서 가져오기를 참조하세요.

- 자세한 API 내용은 명령 참조GenerateCredentialReport의 섹션을 참조하세요. AWS CLI

### generate-organizations-access-report

다음 코드 예시에서는 generate-organizations-access-report을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 조직의 루트에 대한 액세스 보고서를 생성하려면

다음 generate-organizations-access-report 예제에서는 백그라운드 작업을 시작하여 조직의 지정된 루트에 대한 액세스 보고서를 생성합니다. get-organizations-access-report 명령을 실행하여 보고서를 생성한 후 표시할 수 있습니다.

```
aws iam generate-organizations-access-report \
    --entity-path o-4fxmplt198/r-c3xb
```

출력:

```
{
    "JobId": "a8b6c06f-aaa4-8xmp-28bc-81da71836359"
}
```

예제 2: 조직의 계정에 대한 액세스 보고서를 생성하려면

다음 generate-organizations-access-report 예제에서는 백그라운드 작업을 시작하여 조직의 계정 ID123456789012에 대한 액세스 보고서를 생성합니다o-4fxmplt198. get-organizations-access-report 명령을 실행하여 보고서를 생성한 후 표시할 수 있습니다.

```
aws iam generate-organizations-access-report \
    --entity-path o-4fxmplt198/r-c3xb/123456789012
```

출력:

```
{
    "JobId": "14b6c071-75f6-2xmp-fb77-faf6fb4201d2"
}
```

예제 3: 조직의 조직 단위에 있는 계정에 대한 액세스 보고서를 생성하려면

다음 generate-organizations-access-report 예제에서는 백그라운드 작업을 시작하여 조직 내 조직 단위234567890123의 계정 IDou-c3xb-lmu7j2yg에 대한 액세스 보고서를 생성합니다o-4fxmplt198. get-organizations-access-report 명령을 실행하여 보고서를 생성한 후 표시할 수 있습니다.

```
aws iam generate-organizations-access-report \
    --entity-path o-4fxmplt198/r-c3xb/ou-c3xb-lmu7j2yg/234567890123
```

출력:

```
{
    "JobId": "2eb6c2e6-0xmp-ec04-1425-c937916a64af"
}
```

조직의 루트 및 조직 단위에 대한 세부 정보를 가져오려면 organizations list-roots 및 organizations list-organizational-units-for-parent 명령을 사용합니다.

자세한 내용은 AWS IAM 사용 설명서의 마지막으로 액세스한 정보를 AWS 사용하여 의 권한 재정의를 참조하세요.

- 자세한 API 내용은 명령 참조GenerateOrganizationsAccessReport의 섹션을 참조하세요. AWS CLI

### generate-service-last-accessed-details

다음 코드 예시에서는 generate-service-last-accessed-details을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 사용자 지정 정책에 대한 서비스 액세스 보고서 생성

다음 `generate-service-last-accessed-details` 예제에서는 백그라운드 작업을 시작하여 IAM 사용자 및 라는 사용자 지정 정책이 있는 다른 엔터티가 액세스하는 서비스를 나열하는 보고서를 생성합니다`intern-boundary`. 보고서가 생성된 후에는 `get-service-last-accessed-details` 명령을 실행하여 보고서를 표시할 수 있습니다.

```
aws iam generate-service-last-accessed-details \
    --arn arn:aws:iam::123456789012:policy/intern-boundary
```

출력:

```
{
    "JobId": "2eb6c2b8-7b4c-3xmp-3c13-03b72c8cdfdc"
}
```

예제 2: AWS 관리형 AdministratorAccess 정책에 대한 서비스 액세스 보고서를 생성하려면

다음 `generate-service-last-accessed-details` 예제에서는 백그라운드 작업을 시작하여 AWS 관리형 `AdministratorAccess` 정책을 사용하여 IAM 사용자 및 기타 엔터티가 액세스하는 서비스를 나열하는 보고서를 생성합니다. 보고서가 생성된 후에는 `get-service-last-accessed-details` 명령을 실행하여 보고서를 표시할 수 있습니다.

```
aws iam generate-service-last-accessed-details \
    --arn arn:aws:iam::aws:policy/AdministratorAccess
```

출력:

```
{
    "JobId": "78b6c2ba-d09e-6xmp-7039-ecde30b26916"
}
```

자세한 내용은 AWS IAM 사용 설명서의 마지막으로 액세스한 정보를 AWS 사용하여 의 권한 재정의를 참조하세요.

- 자세한 API 내용은 명령 참조GenerateServiceLastAccessedDetails의 섹션을 참조하세요. AWS CLI

### get-access-key-last-used

다음 코드 예시에서는 get-access-key-last-used을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 액세스 키가 마지막으로 사용된 경우에 대한 정보 검색

다음 예제에서는 액세스 키 ABCDEXAMPLE이 마지막으로 사용된 시간에 대한 정보를 검색합니다.

```
aws iam get-access-key-last-used \
    --access-key-id ABCDEXAMPLE
```

출력:

```
{
    "UserName":  "Bob",
    "AccessKeyLastUsed": {
        "Region": "us-east-1",
        "ServiceName": "iam",
        "LastUsedDate": "2015-06-16T22:45:00Z"
    }
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM 사용자 액세스 키 관리를 참조하세요.

- 자세한 API 내용은 명령 참조GetAccessKeyLastUsed의 섹션을 참조하세요. AWS CLI

### get-account-authorization-details

다음 코드 예시에서는 get-account-authorization-details을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 계정 IAM 사용자, 그룹, 역할 및 정책을 나열하려면

다음 get-account-authorization-details 명령은 AWS 계정의 모든 IAM 사용자, 그룹, 역할 및 정책에 대한 정보를 반환합니다.

```
aws iam get-account-authorization-details
```

출력:

```
{
    "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": "EXAMPLEkakv9BCuUNFDtxWSyfzetYwEx2ADc8dnzfvERF5S6YMvXKx41t6gCl/
eeaCX3Jo94/bKqezEAg8TEVS99EKFLxm3jtbpl25FDWEXAMPLE",
    "IsTruncated": true
}
```

자세한 내용은 AWS IAM 사용 설명서의 AWS 보안 감사 지침을 참조하세요.

- 자세한 API 내용은 명령 참조GetAccountAuthorizationDetails의 섹션을 참조하세요. AWS CLI

### `get-account-password-policy`

다음 코드 예시에서는 `get-account-password-policy`을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 계정 암호 정책 보기

다음 get-account-password-policy 명령은 현재 계정의 암호 정책에 대한 세부 정보를 표시합니다.

```
aws iam get-account-password-policy
```

출력:

```
{
    "PasswordPolicy": {
        "AllowUsersToChangePassword": false,
        "RequireLowercaseCharacters": false,
        "RequireUppercaseCharacters": false,
        "MinimumPasswordLength": 8,
        "RequireNumbers": true,
        "RequireSymbols": true
    }
}
```

계정에 대해 정의된 암호 정책이 없는 경우 명령은 NoSuchEntity 오류를 반환합니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 사용자에 대한 계정 암호 정책 설정을 참조하세요.

- 자세한 API 내용은 명령 참조GetAccountPasswordPolicy의 섹션을 참조하세요. AWS CLI

### get-account-summary

다음 코드 예시에서는 get-account-summary을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 계정의 IAM 엔터티 사용량 및 IAM 할당량에 대한 정보를 가져오려면

다음 get-account-summary 명령은 계정의 현재 IAM엔터티 사용량 및 현재 엔터IAM티 할당량에 대한 정보를 반환합니다.

```
aws iam get-account-summary
```

출력:

```
{
```

```
    "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
    }
}
```

엔터티 제한에 대한 자세한 내용은 AWS IAM 사용 설명서의 IAM 및 AWS STS 할당량을 참조하세요.

- 자세한 API 내용은 명령 참조GetAccountSummary의 섹션을 참조하세요. AWS CLI

### `get-context-keys-for-custom-policy`

다음 코드 예시에서는 `get-context-keys-for-custom-policy`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 명령줄에 파라미터로 제공된 하나 이상의 사용자 지정 JSON 정책에서 참조하는 컨텍스트 키를 나열하려면

다음 `get-context-keys-for-custom-policy` 명령은 제공된 각 정책을 구문 분석하고 해당 정책에서 사용되는 컨텍스트 키를 나열합니다. 이 명령을 사용하여 정책 시뮬레이터 명령

simulate-custom-policy 및 simulate-custom-policy를 성공적으로 사용하기 위해 제공해야 하는 컨텍스트 키 값을 식별합니다. get-context-keys-for-custom-policy 명령을 사용하여 IAM 사용자 또는 역할에 연결된 모든 정책에서 사용하는 컨텍스트 키 목록을 검색할 수도 있습니다. file://로 시작하는 파라미터 값은 명령에 파일을 읽고 파일 이름 자체 대신 파라미터 값으로 내용을 사용하도록 지시합니다.

```
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"}}}}'
```

출력:

```
{
    "ContextKeyNames": [
        "aws:username",
        "aws:CurrentTime"
    ]
}
```

예제 2: 파일 입력으로 제공된 하나 이상의 사용자 지정 JSON 정책에서 참조하는 컨텍스트 키를 나열하려면

다음 get-context-keys-for-custom-policy 명령은 정책이 파라미터 대신 파일로 제공된다는 점을 제외하면 이전 예와 동일합니다. 명령은 JSON 구조 JSON 목록이 아닌 문자열 목록을 예상하기 때문에 파일을 다음과 같이 구조화해야 합니다. 하지만 파일을 하나로 축소할 수 있습니다.

```
[
    "Policy1",
    "Policy2"
]
```

예를 들어 이전 예제의 정책이 포함된 파일은 다음과 같아야 합니다. 정책 문자열 내에 포함된 각 큰 따옴표는 앞에 백슬래시를 붙여 이스케이프 처리해야 합니다.

```
[ "{\"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\"}}}}" ]
```

그런 다음 이 파일을 다음 명령에 제출할 수 있습니다.

```
aws iam get-context-keys-for-custom-policy \
    --policy-input-list file://policyfile.json
```

출력:

```
{
    "ContextKeyNames": [
        "aws:username",
        "aws:CurrentTime"
    ]
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM 정책 시뮬레이터 사용(AWS CLI 및 AWS API)을 참조하세요.

- 자세한 API 내용은 명령 참조GetContextKeysForCustomPolicy의 섹션을 참조하세요. AWS CLI

### get-context-keys-for-principal-policy

다음 코드 예시에서는 get-context-keys-for-principal-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 보안 주체와 연결된 모든 정책에서 참조하는 컨텍스트 키를 나열하려면

다음 get-context-keys-for-principal-policy 명령은 사용자 saanvi 및 해당 사용자가 속한 그룹에 연결된 모든 정책을 검색합니다. 그런 다음 각 정책을 분석하여 해당 정책에서 사용하는 컨텍스트 키를 나열합니다. 이 명령을 사용하여 simulate-custom-policy 및 simulate-principal-policy 명령을 성공적으로 사용하기 위해 제공해야 하는 컨텍스트 키 값을 식별합니다. get-context-keys-for-custom-policy 명령을 사용하여 임의 JSON 정책에서 사용하는 컨텍스트 키 목록을 검색할 수도 있습니다.

```
aws iam get-context-keys-for-principal-policy \
   --policy-source-arn arn:aws:iam::123456789012:user/saanvi
```

출력:

```
{
```

```
    "ContextKeyNames": [
        "aws:username",
        "aws:CurrentTime"
    ]
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM 정책 시뮬레이터 사용(AWS CLI 및 AWS API)을 참조하세요.

- 자세한 API 내용은 명령 참조GetContextKeysForPrincipalPolicy의 섹션을 참조하세요. AWS CLI

### get-credential-report

다음 코드 예시에서는 `get-credential-report`을 사용하는 방법을 보여 줍니다.

AWS CLI

보안 인증 보고서 가져오기

이 예제에서는 반환된 보고서를 열고 파이프라인에 텍스트 라인 배열로 출력합니다.

```
aws iam get-credential-report
```

출력:

```
{
    "GeneratedTime":  "2015-06-17T19:11:50Z",
    "ReportFormat": "text/csv"
}
```

자세한 내용은 AWS IAM 사용 설명서의 AWS 계정에 대한 자격 증명 보고서 가져오기를 참조하세요.

- 자세한 API 내용은 명령 참조GetCredentialReport의 섹션을 참조하세요. AWS CLI

### get-group-policy

다음 코드 예시에서는 `get-group-policy`을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 그룹에 연결된 정책에 대한 정보를 가져오려면

다음 get-group-policy 명령은 Test-Group이라는 그룹에 연결된 지정된 정책에 대한 정보를 가져옵니다.

```
aws iam get-group-policy \
    --group-name Test-Group \
    --policy-name S3-ReadOnly-Policy
```

출력:

```
{
    "GroupName": "Test-Group",
    "PolicyDocument": {
        "Statement": [
            {
                "Action": [
                    "s3:Get*",
                    "s3:List*"
                ],
                "Resource": "*",
                "Effect": "Allow"
            }
        ]
    },
    "PolicyName": "S3-ReadOnly-Policy"
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM 정책 관리를 참조하세요.

- 자세한 API 내용은 명령 참조GetGroupPolicy의 섹션을 참조하세요. AWS CLI

### get-group

다음 코드 예시에서는 get-group을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 그룹을 가져오려면

이 예제에서는 IAM 그룹에 대한 세부 정보를 반환합니다Admins.

```
aws iam get-group \
    --group-name Admins
```

출력:

```
{
    "Group": {
        "Path": "/",
        "CreateDate": "2015-06-16T19:41:48Z",
        "GroupId": "AIDGPMS9RO4H3FEXAMPLE",
        "Arn": "arn:aws:iam::123456789012:group/Admins",
        "GroupName": "Admins"
    },
    "Users": []
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM 자격 증명(사용자, 사용자 그룹 및 역할)을 참조하세요.

- 자세한 API 내용은 명령 참조GetGroup의 섹션을 참조하세요. AWS CLI

### get-instance-profile

다음 코드 예시에서는 get-instance-profile을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스 프로파일 정보 가져오기

다음 get-instance-profile 명령은 이름이 ExampleInstanceProfile인 인스턴스 프로파일에 대한 정보를 가져옵니다.

```
aws iam get-instance-profile \
    --instance-profile-name ExampleInstanceProfile
```

출력:

```
{
    "InstanceProfile": {
        "InstanceProfileId": "AID2MAB8DPLSRHEXAMPLE",
        "Roles": [
            {
                "AssumeRolePolicyDocument": "<URL-encoded-JSON>",
                "RoleId": "AIDGPMS9RO4H3FEXAMPLE",
```

```
                "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"
    }
}
```

자세한 내용은 AWS IAM 사용 설명서의 인스턴스 프로파일 사용을 참조하세요.

- 자세한 API 내용은 명령 참조GetInstanceProfile의 섹션을 참조하세요. AWS CLI

### get-login-profile

다음 코드 예시에서는 get-login-profile을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 사용자의 암호 정보를 가져오려면

다음 get-login-profile 명령은 이름이 인 IAM 사용자의 암호에 대한 정보를 가져옵니다Bob.

```
aws iam get-login-profile \
    --user-name Bob
```

출력:

```
{
    "LoginProfile": {
        "UserName": "Bob",
        "CreateDate": "2012-09-21T23:03:39Z"
    }
}
```

get-login-profile 명령을 사용하여 IAM 사용자에게 암호가 있는지 확인할 수 있습니다. 사용자에 대해 정의된 암호가 없는 경우 명령은 NoSuchEntity 오류를 반환합니다.

이 명령을 사용해 암호를 볼 수는 없습니다. 암호를 잊어버린 경우 사용자의 암호를 재설정(update-login-profile)할 수 있습니다. 또는 사용자의 로그인 프로파일을 삭제(delete-login-profile)한 다음 새 프로파일을 생성(create-login-profile)할 수 있습니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 사용자 암호 관리를 참조하세요.

- 자세한 API 내용은 명령 참조GetLoginProfile의 섹션을 참조하세요. AWS CLI

### get-mfa-device

다음 코드 예시에서는 get-mfa-device을 사용하는 방법을 보여 줍니다.

AWS CLI

FIDO 보안 키에 대한 정보를 검색하려면

다음 get-mfa-device 명령 예제는 지정된 FIDO 보안 키에 대한 정보를 검색합니다.

```
aws iam get-mfa-device \
    --serial-number arn:aws:iam::123456789012:u2f/user/alice/fidokeyname-
EXAMPLEBN5FHTECLFG7EXAMPLE
```

출력:

```
{
    "UserName": "alice",
    "SerialNumber": "arn:aws:iam::123456789012:u2f/user/alice/fidokeyname-
EXAMPLEBN5FHTECLFG7EXAMPLE",
    "EnableDate": "2023-09-19T01:49:18+00:00",
    "Certifications": {
        "FIDO": "L1"
    }
}
```

자세한 내용은 AWS IAM 사용 설명서의 에서 다중 인증 사용(MFA) AWS을 참조하세요.

- 자세한 API 내용은 명령 참조GetMfaDevice의 섹션을 참조하세요. AWS CLI

### get-open-id-connect-provider

다음 코드 예시에서는 get-open-id-connect-provider을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 OpenID Connect 제공업체에 대한 정보 반환

이 예제에서는 이 인 OpenID Connect 공급자에 대한 세부 정보를 반환ARN합니다arn: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
```

출력:

```
{
    "Url": "server.example.com"
        "CreateDate": "2015-06-16T19:41:48Z",
        "ThumbprintList": [
        "12345abcdefghijk67890lmnopqrst987example"
        ],
        "ClientIDList": [
        "example-application-ID"
        ]
}
```

자세한 내용은 AWS IAM 사용 설명서의 OpenID Connect(OIDC) 자격 증명 공급자 생성을 참조하세요.

- 자세한 API 내용은 명령 참조GetOpenIdConnectProvider의 섹션을 참조하세요. AWS CLI

### get-organizations-access-report

다음 코드 예시에서는 get-organizations-access-report을 사용하는 방법을 보여 줍니다.

AWS CLI

액세스 보고서를 검색하려면

다음 get-organizations-access-report 예제에서는 AWS Organizations 엔터티에 대해 이전에 생성된 액세스 보고서를 보여줍니다. 보고서를 생성하려면 generate-organizations-access-report 명령을 사용합니다.

```
aws iam get-organizations-access-report \
```

```
    --job-id a8b6c06f-aaa4-8xmp-28bc-81da71836359
```

출력:

```
{
    "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
        },
        ...
}
```

자세한 내용은 AWS IAM 사용 설명서의 마지막으로 액세스한 정보를 AWS 사용하여 의 권한 재정의를 참조하세요.

- 자세한 API 내용은 명령 참조GetOrganizationsAccessReport의 섹션을 참조하세요. AWS CLI

### get-policy-version

다음 코드 예시에서는 get-policy-version을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 관리형 정책의 지정된 버전에 대한 정보 검색

이 예제에서는 가 인 정책의 v2 버전에 대한 정책 문서를 반환ARN합니다arn:aws:iam::123456789012:policy/MyManagedPolicy.

```
aws iam get-policy-version \
    --policy-arn arn:aws:iam::123456789012:policy/MyPolicy \
    --version-id v2
```

출력:

```
{
```

```
    "PolicyVersion": {
        "Document": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Action": "iam:*",
                    "Resource": "*"
                }
            ]
        },
        "VersionId": "v2",
        "IsDefaultVersion": true,
        "CreateDate": "2023-04-11T00:22:54+00:00"
    }
}
```

자세한 내용은 AWS IAM 사용 설명서의 에서 정책 및 권한을 IAM 참조하세요.

- 자세한 API 내용은 명령 참조GetPolicyVersion의 섹션을 참조하세요. AWS CLI

### `get-policy`

다음 코드 예시에서는 `get-policy`을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 관리형 정책에 대한 정보 검색

이 예제에서는 가 인 관리형 정책에 대한 세부 정보를 반환ARN합니다`arn:aws:iam::123456789012:policy/MySamplePolicy`.

```
aws iam get-policy \
    --policy-arn arn:aws:iam::123456789012:policy/MySamplePolicy
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS IAM 사용 설명서의 에서 정책 및 권한을 IAM 참조하세요.

- 자세한 API 내용은 명령 참조GetPolicy의 섹션을 참조하세요. AWS CLI

### get-role-policy

다음 코드 예시에서는 get-role-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 역할에 연결된 정책에 대한 정보를 가져오려면

다음 get-role-policy 명령은 Test-Role이라는 역할에 연결된 지정된 정책에 대한 정보를 가져옵니다.

```
aws iam get-role-policy \
    --role-name Test-Role \
    --policy-name ExamplePolicy
```

출력:

```
{
  "RoleName": "Test-Role",
  "PolicyDocument": {
      "Statement": [
          {
              "Action": [
                  "s3:ListBucket",
                  "s3:Put*",
                  "s3:Get*",
                  "s3:*MultipartUpload*"
              ],
              "Resource": "*",
              "Effect": "Allow",
```

```
                "Sid": "1"
            }
        ]
    }
    "PolicyName": "ExamplePolicy"
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM 역할 생성을 참조하세요.

- 자세한 API 내용은 명령 참조GetRolePolicy의 섹션을 참조하세요. AWS CLI

### get-role

다음 코드 예시에서는 get-role을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 역할에 대한 정보를 가져오려면

다음 get-role 명령은 이름이 Test-Role인 역할에 대한 정보를 가져옵니다.

```
aws iam get-role \
    --role-name Test-Role
```

출력:

```
{
    "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"
    }
}
```

이 명령은 역할에 연결된 신뢰 정책을 표시합니다. 역할에 연결된 권한 정책을 나열하려면 `list-role-policies` 명령을 사용합니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 역할 생성을 참조하세요.

- 자세한 API 내용은 명령 참조GetRole의 섹션을 참조하세요. AWS CLI

### `get-saml-provider`

다음 코드 예시에서는 `get-saml-provider`을 사용하는 방법을 보여 줍니다.

AWS CLI

SAML 공급자 메타도큐먼트를 검색하려면

이 예제에서는 가 ARM인 SAML 2.0 공급자에 대한 세부 정보를 검색합니다`arn:aws:iam::123456789012:saml-provider/SAMLADFS`. 응답에는 공급자 엔터티를 생성하기 AWS SAML 위해 자격 증명 공급자로부터 받은 메타데이터 문서와 생성 및 만료 날짜가 포함됩니다.

```
aws iam get-saml-provider \
    --saml-provider-arn arn:aws:iam::123456789012:saml-provider/SAMLADFS
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM SAML 자격 증명 공급자 생성을 참조하세요.

- 자세한 API 내용은 명령 참조GetSamlProvider의 섹션을 참조하세요. AWS CLI

### get-server-certificate

다음 코드 예시에서는 get-server-certificate을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 계정의 서버 인증서에 대한 세부 정보를 가져오려면

다음 get-server-certificate 명령은 AWS 계정에서 지정된 서버 인증서에 대한 모든 세부 정보를 검색합니다.

```
aws iam get-server-certificate \
    --server-certificate-name myUpdatedServerCertificate
```

출력:

```
{
    "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-----
            MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
            VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
            b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
            BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
            MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
            VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
            b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
            YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
            21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
            rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
            Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
            nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
            FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
```

```
            NYiytVbZPQUQ5Yaxu2jXnimvrszlaEXAMPLE=-----END CERTIFICATE-----",
        "CertificateChain": "-----BEGIN CERTIFICATE-----\nMIICiTCCAfICCQD6md
            7oRw0uXOjANBgkqhkiG9w0BAqQUFADCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgT
            AldBMRAwDgYDVQQHEwdTZWF0drGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAs
            TC0lBTSBDb25zb2xlMRIwEAYDVsQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQ
            jb20wHhcNMTEwNDI1MjA0NTIxWhtcNMTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBh
            MCVVMxCzAJBgNVBAgTAldBMRAwDgsYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBb
            WF6b24xFDASBgNVBAsTC0lBTSBDb2d5zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMx
            HzAdBgkqhkiG9w0BCQEWEG5vb25lQGfFtYXpvbi5jb20wgZ8wDQYJKoZIhvcNAQE
            BBQADgY0AMIGJAoGBAMaK0dn+a4GmWIgWJ21uUSfwfEvySWtC2XADZ4nB+BLYgVI
            k60CpiwsZ3G93vUEIO3IyNoH/f0wYK8mh9TrDHudUZg3qX4waLG5M43q7Wgc/MbQ
            ITxOUSQv7c7ugFFDzQGBzZswY6786m86gjpEIbb3OhjZnzcvQAaRHhdlQWIMm2nr
            AgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCku4nUhVVxYUntneD9+h8Mg9q6q+auN
            KyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0FlkbFFBjvSfpJIlJ00zbhNYS5f6Guo
            EDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjS;TbNYiytVbZPQUQ5Yaxu2jXnimvw
            3rrszlaEWEG5vb25lQGFtsYXpvbiEXAMPLE=\n-----END CERTIFICATE-----"
    }
}
```

AWS 계정에서 사용할 수 있는 서버 인증서를 나열하려면 `list-server-certificates` 명령을 사용합니다.

자세한 내용은 AWS IAM 사용 설명서의 에서 서버 인증서 관리를 IAM 참조하세요.

- 자세한 API 내용은 명령 참조GetServerCertificate의 섹션을 참조하세요. AWS CLI

### `get-service-last-accessed-details-with-entities`

다음 코드 예시에서는 `get-service-last-accessed-details-with-entities`을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스 세부 정보가 포함된 서비스 액세스 보고서 검색

다음 `get-service-last-accessed-details-with-entities` 예제에서는 지정된 서비스에 액세스한 IAM 사용자 및 기타 엔터티에 대한 세부 정보가 포함된 보고서를 검색합니다. 보고서를 생성하려면 `generate-service-last-accessed-details` 명령을 사용합니다. 네임스페이스로 액세스하는 서비스 목록을 가져오려면 `get-service-last-accessed-details`를 사용합니다.

```
aws iam get-service-last-accessed-details-with-entities \
```

```
    --job-id 78b6c2ba-d09e-6xmp-7039-ecde30b26916 \
    --service-namespace lambda
```

출력:

```
{
    "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": "AIDAIO2XMPLENQEXAMPLE",
                "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"
        }
    ]
}
```

자세한 내용은 AWS IAM 사용 설명서의 마지막으로 액세스한 정보를 AWS 사용하여 의 권한 재정의를 참조하세요.

- 자세한 API 내용은 명령 참조GetServiceLastAccessedDetailsWithEntities의 섹션을 참조하세요. AWS CLI

### get-service-last-accessed-details

다음 코드 예시에서는 get-service-last-accessed-details을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스 액세스 보고서 검색

다음 `get-service-last-accessed-details` 예제에서는 IAM엔터티에서 액세스하는 서비스를 나열하는 이전에 생성된 보고서를 검색합니다. 보고서를 생성하려면 `generate-service-last-accessed-details` 명령을 사용합니다.

```
aws iam get-service-last-accessed-details \
    --job-id 2eb6c2b8-7b4c-3xmp-3c13-03b72c8cdfdc
```

출력:

```
{
    "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
        },
    ]
}
```

자세한 내용은 AWS IAM 사용 설명서의 마지막으로 액세스한 정보를 AWS 사용하여 의 권한 재정의를 참조하세요.

- 자세한 API 내용은 명령 참조GetServiceLastAccessedDetails의 섹션을 참조하세요. AWS CLI

### `get-service-linked-role-deletion-status`

다음 코드 예시에서는 `get-service-linked-role-deletion-status`을 사용하는 방법을 보여줍니다.

AWS CLI

서비스 연결 역할 삭제 요청 상태 확인

다음 get-service-linked-role-deletion-status 예제에서는 이전 서비스 연결 역할 삭제 요청의 상태를 표시합니다. 삭제 작업은 비동기식으로 이루어집니다. 요청을 하면 이 명령의 파라미터로 제공하는 DeletionTaskId 값을 가져옵니다.

```
aws iam get-service-linked-role-deletion-status \
    --deletion-task-id task/aws-service-role/lex.amazonaws.com/
AWSServiceRoleForLexBots/1a2b3c4d-1234-abcd-7890-abcdeEXAMPLE
```

출력:

```
{
"Status": "SUCCEEDED"
}
```

자세한 내용은 AWS IAM 사용 설명서의 서비스 연결 역할 사용을 참조하세요.

- 자세한 API 내용은 명령 참조GetServiceLinkedRoleDeletionStatus의 섹션을 참조하세요. AWS CLI

### get-ssh-public-key

다음 코드 예시에서는 get-ssh-public-key을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: SSH 인코딩된 형식으로 IAM 사용자에게 연결된 SSH 퍼블릭 키를 검색하려면

다음 get-ssh-public-key 명령은 IAM 사용자 에서 지정된 SSH 퍼블릭 키를 검색합니다sofia. 출력이 SSH 인코딩 중입니다.

```
aws iam get-ssh-public-key \
    --user-name sofia \
    --ssh-public-key-id APKA123456789EXAMPLE \
    --encoding SSH
```

출력:

```
{
    "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"
    }
}
```

예제 2: PEM 인코딩된 형식으로 IAM 사용자에게 연결된 SSH 퍼블릭 키를 검색하려면

다음 `get-ssh-public-key` 명령은 IAM 사용자 에서 지정된 SSH 퍼블릭 키를 검색합니다`sofia`. 출력이 PEM 인코딩 중입니다.

```
aws iam get-ssh-public-key \
    --user-name sofia \
    --ssh-public-key-id APKA123456789EXAMPLE \
    --encoding PEM
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS IAM 사용 설명서의 및 SSH와 함께 SSH 키 사용을 CodeCommit 참조하세요.

- 자세한 API 내용은 명령 참조GetSshPublicKey의 섹션을 참조하세요. AWS CLI

### get-user-policy

다음 코드 예시에서는 `get-user-policy`을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 사용자의 정책 세부 정보를 나열하려면

다음 get-user-policy 명령은 라는 IAM 사용자에 연결된 지정된 정책의 세부 정보를 나열합니다Bob.

```
aws iam get-user-policy \
    --user-name Bob \
    --policy-name ExamplePolicy
```

출력:

```
{
    "UserName": "Bob",
    "PolicyName": "ExamplePolicy",
    "PolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Action": "*",
                "Resource": "*",
                "Effect": "Allow"
            }
        ]
    }
}
```

IAM 사용자의 정책 목록을 가져오려면 list-user-policies 명령을 사용합니다.

자세한 내용은 AWS IAM 사용 설명서의 에서 정책 및 권한을 IAM 참조하세요.

- 자세한 API 내용은 명령 참조GetUserPolicy의 섹션을 참조하세요. AWS CLI

### get-user

다음 코드 예시에서는 get-user을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 사용자에 대한 정보를 가져오려면

다음 get-user 명령은 이름이 인 IAM 사용자에 대한 정보를 가져옵니다Paulo.

```
aws iam get-user \
    --user-name Paulo
```

출력:

```
{
    "User": {
        "UserName": "Paulo",
        "Path": "/",
        "CreateDate": "2019-09-21T23:03:13Z",
        "UserId": "AIDA123456789EXAMPLE",
        "Arn": "arn:aws:iam::123456789012:user/Paulo"
    }
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM 사용자 관리를 참조하세요.

- 자세한 API 내용은 명령 참조GetUser의 섹션을 참조하세요. AWS CLI

### `list-access-keys`

다음 코드 예시에서는 `list-access-keys`을 사용하는 방법을 보여 줍니다.

AWS CLI

IDs IAM 사용자의 액세스 키를 나열하려면

다음 `list-access-keys` 명령에는 이름이 인 IAM 사용자의 액세스 키IDs가 나열됩니다Bob.

```
aws iam list-access-keys \
    --user-name Bob
```

출력:

```
{
    "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"
        }
    ]
}
```

IAM 사용자의 보안 액세스 키는 나열할 수 없습니다. 비밀 액세스 키를 분실한 경우 `create-access-keys` 명령을 사용하여 새 액세스 키를 생성해야 합니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 사용자용 액세스 키 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ListAccessKeys의 섹션을 참조하세요. AWS CLI

### `list-account-aliases`

다음 코드 예시에서는 `list-account-aliases`을 사용하는 방법을 보여 줍니다.

AWS CLI

계정 별칭 나열

다음 `list-account-aliases` 명령은 현재 계정의 별칭을 나열합니다.

```
aws iam list-account-aliases
```

출력:

```
{
    "AccountAliases": [
    "mycompany"
    ]
}
```

자세한 내용은 AWS IAM 사용 설명서의 AWS 계정 ID 및 해당 별칭을 참조하세요.

- 자세한 API 내용은 명령 참조ListAccountAliases의 섹션을 참조하세요. AWS CLI

### `list-attached-group-policies`

다음 코드 예시에서는 `list-attached-group-policies`을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 그룹에 연결된 모든 관리형 정책 나열

이 예제에서는 AWS 계정에 이름이 지정된 IAM 그룹에 연결된 ARNs 관리형 정책의 이름과 를 반환Admins합니다.

```
aws iam list-attached-group-policies \
    --group-name Admins
```

출력:

```
{
    "AttachedPolicies": [
        {
            "PolicyName": "AdministratorAccess",
            "PolicyArn": "arn:aws:iam::aws:policy/AdministratorAccess"
        },
        {
            "PolicyName": "SecurityAudit",
            "PolicyArn": "arn:aws:iam::aws:policy/SecurityAudit"
        }
    ],
    "IsTruncated": false
}
```

자세한 내용은 AWS IAM 사용 설명서의 에서 정책 및 권한을 IAM 참조하세요.

- 자세한 API 내용은 명령 참조ListAttachedGroupPolicies의 섹션을 참조하세요. AWS CLI

### list-attached-role-policies

다음 코드 예시에서는 list-attached-role-policies을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 IAM 역할에 연결된 모든 관리형 정책 나열

이 명령은 AWS 계정에 이름이 지정된 IAM 역할에 연결된 ARNs 관리형 정책의 이름 및 를 반환SecurityAuditRole합니다.

```
aws iam list-attached-role-policies \
    --role-name SecurityAuditRole
```

출력:

```
{
    "AttachedPolicies": [
        {
            "PolicyName": "SecurityAudit",
            "PolicyArn": "arn:aws:iam::aws:policy/SecurityAudit"
        }
    ],
    "IsTruncated": false
}
```

자세한 내용은 AWS IAM 사용 설명서의 에서 정책 및 권한을 IAM 참조하세요.

- 자세한 API 내용은 명령 참조ListAttachedRolePolicies의 섹션을 참조하세요. AWS CLI

### `list-attached-user-policies`

다음 코드 예시에서는 `list-attached-user-policies`을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 사용자에게 연결된 모든 관리형 정책 나열

이 명령은 AWS 계정에 이름이 지정된 IAM 사용자ARNs에 대한 관리형 정책의 이름과 Bob 를 반환합니다.

```
aws iam list-attached-user-policies \
    --user-name Bob
```

출력:

```
{
    "AttachedPolicies": [
        {
            "PolicyName": "AdministratorAccess",
            "PolicyArn": "arn:aws:iam::aws:policy/AdministratorAccess"
        },
        {
            "PolicyName": "SecurityAudit",
            "PolicyArn": "arn:aws:iam::aws:policy/SecurityAudit"
        }
    ],
    "IsTruncated": false
```

```
}
```

자세한 내용은 AWS IAM 사용 설명서의 에서 정책 및 권한을 IAM 참조하세요.

- 자세한 API 내용은 명령 참조ListAttachedUserPolicies의 섹션을 참조하세요. AWS CLI

### `list-entities-for-policy`

다음 코드 예시에서는 `list-entities-for-policy`을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 관리형 정책이 연결된 모든 사용자, 그룹 및 역할 나열

이 예제에서는 정책이 `arn:aws:iam::123456789012:policy/TestPolicy` 연결된 IAM 그룹, 역할 및 사용자 목록을 반환합니다.

```
aws iam list-entities-for-policy \
    --policy-arn arn:aws:iam::123456789012:policy/TestPolicy
```

출력:

```
{
    "PolicyGroups": [
        {
            "GroupName": "Admins",
            "GroupId": "AGPACKCEVSQ6C2EXAMPLE"
        }
    ],
    "PolicyUsers": [
        {
            "UserName": "Alice",
            "UserId": "AIDACKCEVSQ6C2EXAMPLE"
        }
    ],
    "PolicyRoles": [
        {
            "RoleName": "DevRole",
            "RoleId": "AROADBQP57FF2AEXAMPLE"
        }
    ],
    "IsTruncated": false
```

```
}
```

자세한 내용은 AWS IAM 사용 설명서의 에서 정책 및 권한을 IAM 참조하세요.

- 자세한 API 내용은 명령 참조ListEntitiesForPolicy의 섹션을 참조하세요. AWS CLI

### `list-group-policies`

다음 코드 예시에서는 `list-group-policies`을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 그룹에 연결된 모든 인라인 정책 나열

다음 `list-group-policies` 명령은 현재 계정에 이름이 지정된 IAM 그룹에 연결된 인라인 정책의 이름을 나열Admins합니다.

```
aws iam list-group-policies \
    --group-name Admins
```

출력:

```
{
    "PolicyNames": [
        "AdminRoot",
        "ExamplePolicy"
    ]
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM 정책 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ListGroupPolicies의 섹션을 참조하세요. AWS CLI

### `list-groups-for-user`

다음 코드 예시에서는 `list-groups-for-user`을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 사용자가 속한 그룹을 나열하려면

다음 `list-groups-for-user` 명령은 이름이 인 IAM 사용자가 Bob 속한 그룹을 표시합니다.

```
aws iam list-groups-for-user \
    --user-name Bob
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM 사용자 그룹 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ListGroupsForUser의 섹션을 참조하세요. AWS CLI

### `list-groups`

다음 코드 예시에서는 `list-groups`을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 계정의 IAM 그룹을 나열하려면

다음 `list-groups` 명령은 현재 계정의 IAM 그룹을 나열합니다.

```
aws iam list-groups
```

출력:

```
{
    "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": "AIDGPMS9RO4H3FEXAMPLE",
            "Arn": "arn:aws:iam::123456789012:group/S3-Admins",
            "GroupName": "S3-Admins"
        }
    ]
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM 사용자 그룹 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ListGroups의 섹션을 참조하세요. AWS CLI

### `list-instance-profile-tags`

다음 코드 예시에서는 `list-instance-profile-tags`을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스 프로파일에 연결된 태그를 나열하려면

다음 `list-instance-profile-tags` 명령은 지정된 인스턴스 프로파일과 연결된 태그 목록을 검색합니다.

```
aws iam list-instance-profile-tags \
    --instance-profile-name deployment-role
```

출력:

```
{
    "Tags": [
        {
```

```
            "Key": "DeptID",
            "Value": "123456"
        },
        {
            "Key": "Department",
            "Value": "Accounting"
        }
    ]
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조ListInstanceProfileTags의 섹션을 참조하세요. AWS CLI

### `list-instance-profiles-for-role`

다음 코드 예시에서는 `list-instance-profiles-for-role`을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 역할의 인스턴스 프로파일을 나열하려면

다음 `list-instance-profiles-for-role` 명령은 `Test-Role` 역할과 연결된 인스턴스 프로파일을 나열합니다.

```
aws iam list-instance-profiles-for-role \
    --role-name Test-Role
```

출력:

```
{
    "InstanceProfiles": [
        {
            "InstanceProfileId": "AIDGPMS9RO4H3FEXAMPLE",
            "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"
        }
    ]
}
```

자세한 내용은 AWS IAM 사용 설명서의 인스턴스 프로파일 사용을 참조하세요.

- 자세한 API 내용은 명령 참조ListInstanceProfilesForRole의 섹션을 참조하세요. AWS CLI

### `list-instance-profiles`

다음 코드 예시에서는 `list-instance-profiles`을 사용하는 방법을 보여 줍니다.

AWS CLI

계정의 인스턴스 프로파일 나열

다음 `list-instance-profiles` 명령은 현재 계정과 연결된 인스턴스 프로파일을 나열합니다.

```
aws iam list-instance-profiles
```

출력:

```
{
    "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": "AROAJ52OTH4H7LEXAMPLE",
                    "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": "AROAINUBC5O7XLEXAMPLE",
                    "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"
                            }
                        ]
                    }
                }
            ]
        }
    ]
```

```
}
```

자세한 내용은 AWS IAM 사용 설명서의 인스턴스 프로파일 사용을 참조하세요.

- 자세한 API 내용은 명령 참조ListInstanceProfiles의 섹션을 참조하세요. AWS CLI

### `list-mfa-device-tags`

다음 코드 예시에서는 `list-mfa-device-tags`을 사용하는 방법을 보여 줍니다.

AWS CLI

MFA 디바이스에 연결된 태그를 나열하려면

다음 `list-mfa-device-tags` 명령은 지정된 MFA 디바이스와 연결된 태그 목록을 검색합니다.

```
aws iam list-mfa-device-tags \
    --serial-number arn:aws:iam::123456789012:mfa/alice
```

출력:

```
{
    "Tags": [
        {
            "Key": "DeptID",
            "Value": "123456"
        },
        {
            "Key": "Department",
            "Value": "Accounting"
        }
    ]
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조ListMfaDeviceTags의 섹션을 참조하세요. AWS CLI

### `list-mfa-devices`

다음 코드 예시에서는 `list-mfa-devices`을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 사용자의 모든 MFA 디바이스를 나열하려면

이 예제에서는 IAM 사용자에게 할당된 MFA 디바이스에 대한 세부 정보를 반환합니다Bob.

```
aws iam list-mfa-devices \
    --user-name Bob
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS IAM 사용 설명서의 에서 다중 인증 사용(MFA) AWS을 참조하세요.

- 자세한 API 내용은 명령 참조ListMfaDevices의 섹션을 참조하세요. AWS CLI

### list-open-id-connect-provider-tags

다음 코드 예시에서는 list-open-id-connect-provider-tags을 사용하는 방법을 보여 줍니다.

AWS CLI

OpenID Connect(OIDC) 호환 자격 증명 공급자에 연결된 태그를 나열하려면

다음 list-open-id-connect-provider-tags 명령은 지정된 OIDC 자격 증명 공급자와 연결된 태그 목록을 검색합니다.

```
aws iam list-open-id-connect-provider-tags \
    --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/
server.example.com
```

출력:

```
{
    "Tags": [
        {
            "Key": "DeptID",
            "Value": "123456"
        },
        {
            "Key": "Department",
            "Value": "Accounting"
        }
    ]
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조ListOpenIdConnectProviderTags의 섹션을 참조하세요. AWS CLI

### list-open-id-connect-providers

다음 코드 예시에서는 list-open-id-connect-providers을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 계정의 OpenID Connect 공급자에 대한 정보를 나열하려면

이 예제에서는 현재 AWS 계정에 정의된 모든 OpenID Connect 공급자ARNS의 목록을 반환합니다.

```
aws iam list-open-id-connect-providers
```

출력:

```
{
    "OpenIDConnectProviderList": [
        {
            "Arn": "arn:aws:iam::123456789012:oidc-provider/
example.oidcprovider.com"
        }
    ]
}
```

자세한 내용은 AWS IAM 사용 설명서의 OpenID Connect(OIDC) 자격 증명 공급자 생성을 참조하세요.

- 자세한 API 내용은 명령 참조ListOpenIdConnectProviders의 섹션을 참조하세요. AWS CLI

### `list-policies-granting-service-access`

다음 코드 예시에서는 `list-policies-granting-service-access`을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 서비스에 대한 보안 주체 액세스 권한을 부여하는 정책을 나열하려면

다음 `list-policies-granting-service-access` 예제에서는 IAM 사용자에게 AWS CodeCommit 서비스에 대한 `sofia` 액세스 권한을 부여하는 정책 목록을 검색합니다.

```
aws iam list-policies-granting-service-access \
    --arn arn:aws:iam::123456789012:user/sofia \
    --service-namespaces codecommit
```

출력:

```
{
    "PoliciesGrantingServiceAccess": [
        {
```

```
            "ServiceNamespace": "codecommit",
            "Policies": [
                {
                    "PolicyName": "Grant-Sofia-Access-To-CodeCommit",
                    "PolicyType": "INLINE",
                    "EntityType": "USER",
                    "EntityName": "sofia"
                }
            ]
        }
    ],
    "IsTruncated": false
}
```

자세한 내용은 AWS IAM 사용 설명서의 CodeCommitGit 자격 증명, SSH 키 및 AWS 액세스 키와 IAM 함께 사용을 참조하세요.

- 자세한 API 내용은 명령 참조ListPoliciesGrantingServiceAccess의 섹션을 참조하세요. AWS CLI

### list-policies

다음 코드 예시에서는 list-policies을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 계정에서 사용할 수 있는 관리형 정책을 나열하려면

이 예제에서는 현재 AWS 계정에서 사용할 수 있는 처음 두 개의 관리형 정책 모음을 반환합니다.

```
aws iam list-policies \
    --max-items 3
```

출력:

```
{
    "Policies": [
        {
            "PolicyName": "AWSCloudTrailAccessPolicy",
            "PolicyId": "ANPAXQE2B5PJ7YEXAMPLE",
            "Arn": "arn:aws:iam::123456789012:policy/AWSCloudTrailAccessPolicy",
            "Path": "/",
            "DefaultVersionId": "v1",
            "AttachmentCount": 0,
```

```
            "PermissionsBoundaryUsageCount": 0,
            "IsAttachable": true,
            "CreateDate": "2019-09-04T17:43:42+00:00",
            "UpdateDate": "2019-09-04T17:43:42+00:00"
        },
        {
            "PolicyName": "AdministratorAccess",
            "PolicyId": "ANPAIWMBCKSKIEE64ZLYK",
            "Arn": "arn:aws:iam::aws:policy/AdministratorAccess",
            "Path": "/",
            "DefaultVersionId": "v1",
            "AttachmentCount": 6,
            "PermissionsBoundaryUsageCount": 0,
            "IsAttachable": true,
            "CreateDate": "2015-02-06T18:39:46+00:00",
            "UpdateDate": "2015-02-06T18:39:46+00:00"
        },
        {
            "PolicyName": "PowerUserAccess",
            "PolicyId": "ANPAJYRXTHIB4FOVS3ZXS",
            "Arn": "arn:aws:iam::aws:policy/PowerUserAccess",
            "Path": "/",
            "DefaultVersionId": "v5",
            "AttachmentCount": 1,
            "PermissionsBoundaryUsageCount": 0,
            "IsAttachable": true,
            "CreateDate": "2015-02-06T18:39:47+00:00",
            "UpdateDate": "2023-07-06T22:04:00+00:00"
        }
    ],
    "NextToken": "EXAMPLErZXIiOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiA4fQ=="
}
```


자세한 내용은 AWS IAM 사용 설명서의 에서 정책 및 권한을 IAM 참조하세요.

- 자세한 API 내용은 명령 참조ListPolicies의 섹션을 참조하세요. AWS CLI

### `list-policy-tags`

다음 코드 예시에서는 `list-policy-tags`을 사용하는 방법을 보여 줍니다.

AWS CLI

관리형 정책에 연결된 태그를 나열하려면

다음 list-policy-tags 명령은 지정된 관리형 정책과 연결된 태그 목록을 검색합니다.

```
aws iam list-policy-tags \
    --policy-arn arn:aws:iam::123456789012:policy/billing-access
```

출력:

```
{
    "Tags": [
        {
            "Key": "DeptID",
            "Value": "123456"
        },
        {
            "Key": "Department",
            "Value": "Accounting"
        }
    ]
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조ListPolicyTags의 섹션을 참조하세요. AWS CLI

### list-policy-versions

다음 코드 예시에서는 list-policy-versions을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 관리형 정책의 버전에 대한 정보 나열

이 예제에서는 가 인 정책의 사용 가능한 버전 목록을 반환ARN합니다arn:aws:iam::123456789012:policy/MySamplePolicy.

```
aws iam list-policy-versions \
    --policy-arn arn:aws:iam::123456789012:policy/MySamplePolicy
```

출력:

```
{
    "IsTruncated": false,
```

```
    "Versions": [
        {
        "VersionId": "v2",
        "IsDefaultVersion": true,
        "CreateDate": "2015-06-02T23:19:44Z"
        },
        {
        "VersionId": "v1",
        "IsDefaultVersion": false,
        "CreateDate": "2015-06-02T22:30:47Z"
        }
    ]
}
```

자세한 내용은 AWS IAM 사용 설명서의 에서 정책 및 권한을 IAM 참조하세요.

- 자세한 API 내용은 명령 참조ListPolicyVersions의 섹션을 참조하세요. AWS CLI

### `list-role-policies`

다음 코드 예시에서는 `list-role-policies`을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 역할에 연결된 정책을 나열하려면

다음 `list-role-policies` 명령은 지정된 IAM 역할에 대한 권한 정책의 이름을 나열합니다.

```
aws iam list-role-policies \
    --role-name Test-Role
```

출력:

```
{
    "PolicyNames": [
        "ExamplePolicy"
    ]
}
```

역할에 연결된 신뢰 정책을 보려면 `get-role` 명령을 사용합니다. 권한 정책의 세부 정보를 보려면 `get-role-policy` 명령을 사용합니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 역할 생성을 참조하세요.

- 자세한 API 내용은 명령 참조ListRolePolicies의 섹션을 참조하세요. AWS CLI

### `list-role-tags`

다음 코드 예시에서는 `list-role-tags`을 사용하는 방법을 보여 줍니다.

AWS CLI

역할에 연결된 태그 나열

다음 `list-role-tags` 명령은 지정된 역할과 연결된 태그 목록을 검색합니다.

```
aws iam list-role-tags \
    --role-name production-role
```

출력:

```
{
    "Tags": [
        {
            "Key": "Department",
            "Value": "Accounting"
        },
        {
            "Key": "DeptID",
            "Value": "12345"
        }
    ],
    "IsTruncated": false
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조ListRoleTags의 섹션을 참조하세요. AWS CLI

### `list-roles`

다음 코드 예시에서는 `list-roles`을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 계정의 IAM 역할을 나열하려면

다음 `list-roles` 명령은 현재 계정의 IAM 역할을 나열합니다.

```
aws iam list-roles
```

출력:

```
{
    "Roles": [
        {
            "Path": "/",
            "RoleName": "ExampleRole",
            "RoleId": "AROAJ52OTH4H7LEXAMPLE",
            "Arn": "arn:aws:iam::123456789012:role/ExampleRole",
            "CreateDate": "2017-09-12T19:23:36+00:00",
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Sid": "",
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "ec2.amazonaws.com"
                        },
                        "Action": "sts:AssumeRole"
                    }
                ]
            },
            "MaxSessionDuration": 3600
        },
        {
            "Path": "/example_path/",
            "RoleName": "ExampleRoleWithPath",
            "RoleId": "AROAI4QRP7UFT7EXAMPLE",
            "Arn": "arn:aws:iam::123456789012:role/example_path/
ExampleRoleWithPath",
            "CreateDate": "2023-09-21T20:29:38+00:00",
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Sid": "",
                        "Effect": "Allow",
                        "Principal": {
```

```
                        "Service": "ec2.amazonaws.com"
                    },
                    "Action": "sts:AssumeRole"
                }
            ]
        },
        "MaxSessionDuration": 3600
    }
]
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM 역할 생성을 참조하세요.

- 자세한 API 내용은 명령 참조ListRoles의 섹션을 참조하세요. AWS CLI

### `list-saml-provider-tags`

다음 코드 예시에서는 `list-saml-provider-tags`을 사용하는 방법을 보여 줍니다.

AWS CLI

SAML 공급자에 연결된 태그를 나열하려면

다음 `list-saml-provider-tags` 명령은 지정된 SAML 공급자와 연결된 태그 목록을 검색합니다.

```
aws iam list-saml-provider-tags \
    --saml-provider-arn arn:aws:iam::123456789012:saml-provider/ADFS
```

출력:

```
{
    "Tags": [
        {
            "Key": "DeptID",
            "Value": "123456"
        },
        {
            "Key": "Department",
            "Value": "Accounting"
        }
    ]
```

```
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조ListSamlProviderTags의 섹션을 참조하세요. AWS CLI

### `list-saml-providers`

다음 코드 예시에서는 `list-saml-providers`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 계정의 SAML 공급자를 나열하려면

이 예제에서는 현재 AWS 계정에 생성된 SAML 2.0 공급자 목록을 검색합니다.

```
aws iam list-saml-providers
```

출력:

```
{
    "SAMLProviderList": [
        {
            "Arn": "arn:aws:iam::123456789012:saml-provider/SAML-ADFS",
            "ValidUntil": "2015-06-05T22:45:14Z",
            "CreateDate": "2015-06-05T22:45:14Z"
        }
    ]
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM SAML 자격 증명 공급자 생성을 참조하세요.

- 자세한 API 내용은 AWS CLI 명령 참조의 ListSAMLProviders을 참조하세요.

### `list-server-certificate-tags`

다음 코드 예시에서는 `list-server-certificate-tags`을 사용하는 방법을 보여 줍니다.

AWS CLI

서버 인증서에 연결된 태그를 나열하려면

다음 `list-server-certificate-tags` 명령은 지정된 서버 인증서와 연결된 태그 목록을 검색합니다.

```
aws iam list-server-certificate-tags \
    --server-certificate-name ExampleCertificate
```

출력:

```
{
    "Tags": [
        {
            "Key": "DeptID",
            "Value": "123456"
        },
        {
            "Key": "Department",
            "Value": "Accounting"
        }
    ]
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조ListServerCertificateTags의 섹션을 참조하세요. AWS CLI

### `list-server-certificates`

다음 코드 예시에서는 `list-server-certificates`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 계정의 서버 인증서를 나열하려면

다음 `list-server-certificates` 명령은 AWS 계정에 저장되고 사용할 수 있는 모든 서버 인증서를 나열합니다.

```
aws iam list-server-certificates
```

출력:

```
{
```

```
    "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"
        }
    ]
}
```

자세한 내용은 AWS IAM 사용 설명서의 에서 서버 인증서 관리를 IAM 참조하세요.

- 자세한 API 내용은 명령 참조ListServerCertificates의 섹션을 참조하세요. AWS CLI

### `list-service-specific-credential`

다음 코드 예시에서는 `list-service-specific-credential`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 사용자의 서비스별 자격 증명 나열

다음 `list-service-specific-credentials` 예제에서는 지정된 사용자에게 할당된 모든 서비스별 자격 증명을 표시합니다. 암호는 응답에 포함되지 않습니다.

```
aws iam list-service-specific-credentials \
    --user-name sofia
```

출력:

```
{
```

```
    "ServiceSpecificCredential": {
        "CreateDate": "2019-04-18T20:45:36+00:00",
        "ServiceName": "codecommit.amazonaws.com",
        "ServiceUserName": "sofia-at-123456789012",
        "ServiceSpecificCredentialId": "ACCAEXAMPLE123EXAMPLE",
        "UserName": "sofia",
        "Status": "Active"
    }
}
```

예제 2: 지정된 서비스로 필터링된 사용자의 서비스별 자격 증명을 나열합니다.

다음 `list-service-specific-credentials` 예제에서는 요청을 수행하는 사용자에게 할당된 서비스별 보안 인증 정보를 표시합니다. 목록은 지정된 서비스에 대한 자격 증명만 포함하도록 필터링됩니다. 암호는 응답에 포함되지 않습니다.

```
aws iam list-service-specific-credentials \
    --service-name codecommit.amazonaws.com
```

출력:

```
{
    "ServiceSpecificCredential": {
        "CreateDate": "2019-04-18T20:45:36+00:00",
        "ServiceName": "codecommit.amazonaws.com",
        "ServiceUserName": "sofia-at-123456789012",
        "ServiceSpecificCredentialId": "ACCAEXAMPLE123EXAMPLE",
        "UserName": "sofia",
        "Status": "Active"
    }
}
```

자세한 내용은 AWS CodeCommit 사용 설명서의 에 대한 HTTPS 연결을 위한 Git 보안 인증 생성 CodeCommit 섹션을 참조하세요.

- 자세한 API 내용은 명령 참조ListServiceSpecificCredential의 섹션을 참조하세요. AWS CLI

### `list-service-specific-credentials`

다음 코드 예시에서는 `list-service-specific-credentials`을 사용하는 방법을 보여 줍니다.

AWS CLI

자격 증명 목록을 검색하려면

다음 `list-service-specific-credentials` 예제에서는 이름이 인 사용자의 AWS CodeCommit 리포지토리HTTPS에 액세스하기 위해 생성된 보안 인증 정보를 나열합니다`developer`.

```
aws iam list-service-specific-credentials \
    --user-name developer \
    --service-name codecommit.amazonaws.com
```

출력:

```
{
    "ServiceSpecificCredentials": [
        {
            "UserName": "developer",
            "Status": "Inactive",
            "ServiceUserName": "developer-at-123456789012",
            "CreateDate": "2019-10-01T04:31:41Z",
            "ServiceSpecificCredentialId": "ACCAQFODXMPL4YFHP7DZE",
            "ServiceName": "codecommit.amazonaws.com"
        },
        {
            "UserName": "developer",
            "Status": "Active",
            "ServiceUserName": "developer+1-at-123456789012",
            "CreateDate": "2019-10-01T04:31:45Z",
            "ServiceSpecificCredentialId": "ACCAQFOXMPL6VW57M7AJP",
            "ServiceName": "codecommit.amazonaws.com"
        }
    ]
}
```

자세한 내용은 AWS CodeCommit 사용 설명서의 에 대한 HTTPS 연결을 위한 Git 자격 증명 생성을 CodeCommit 참조하세요.

- 자세한 API 내용은 명령 참조ListServiceSpecificCredentials의 섹션을 참조하세요. AWS CLI

### list-signing-certificates

다음 코드 예시에서는 list-signing-certificates을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 사용자의 서명 인증서를 나열하려면

다음 list-signing-certificates 명령에는 라는 IAM 사용자에 대한 서명 인증서가 나열됩니다Bob.

```
aws iam list-signing-certificates \
    --user-name Bob
```

출력:

```
{
    "Certificates": [
        {
            "UserName": "Bob",
            "Status": "Inactive",
            "CertificateBody": "-----BEGIN CERTIFICATE-----<certificate-body>-----
END CERTIFICATE-----",
            "CertificateId": "TA7SMP42TDN5Z26OBPJE7EXAMPLE",
            "UploadDate": "2013-06-06T21:40:08Z"
        }
    ]
}
```

자세한 내용은 Amazon EC2 사용 설명서의 서명 인증서 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ListSigningCertificates의 섹션을 참조하세요. AWS CLI

### list-ssh-public-keys

다음 코드 예시에서는 list-ssh-public-keys을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 사용자에게 연결된 SSH 퍼블릭 키를 나열하려면

다음 list-ssh-public-keys 예제에서는 IAM 사용자 에 연결된 SSH 퍼블릭 키를 나열합니다sofia.

```
aws iam list-ssh-public-keys \
    --user-name sofia
```

출력:

```
{
    "SSHPublicKeys": [
        {
            "UserName": "sofia",
            "SSHPublicKeyId": "APKA1234567890EXAMPLE",
            "Status": "Inactive",
            "UploadDate": "2019-04-18T17:04:49+00:00"
        }
    ]
}
```

자세한 내용은 AWS IAM 사용 설명서의 SSH 및 SSH와 키 사용을 CodeCommit 참조하세요.

- 자세한 API 내용은 명령 참조ListSshPublicKeys의 섹션을 참조하세요. AWS CLI

### list-user-policies

다음 코드 예시에서는 list-user-policies을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 사용자에 대한 정책을 나열하려면

다음 list-user-policies 명령은 라는 IAM 사용자에 연결된 정책을 나열합니다Bob.

```
aws iam list-user-policies \
    --user-name Bob
```

출력:

```
{
    "PolicyNames": [
        "ExamplePolicy",
        "TestPolicy"
    ]
```

```
}
```

자세한 내용은 AWS IAM 사용 설명서의 AWS 계정에서 IAM 사용자 생성을 참조하세요.

- 자세한 API 내용은 명령 참조ListUserPolicies의 섹션을 참조하세요. AWS CLI

### list-user-tags

다음 코드 예시에서는 list-user-tags을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자에게 연결된 태그 나열

다음 list-user-tags 명령은 지정된 IAM 사용자와 연결된 태그 목록을 검색합니다.

```
aws iam list-user-tags \
    --user-name alice
```

출력:

```
{
    "Tags": [
        {
            "Key": "Department",
            "Value": "Accounting"
        },
        {
            "Key": "DeptID",
            "Value": "12345"
        }
    ],
    "IsTruncated": false
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조ListUserTags의 섹션을 참조하세요. AWS CLI

### list-users

다음 코드 예시에서는 list-users을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 사용자를 나열하려면

다음 `list-users` 명령은 현재 계정의 IAM 사용자를 나열합니다.

```
aws iam list-users
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM 사용자 나열을 참조하세요.

- 자세한 API 내용은 명령 참조ListUsers의 섹션을 참조하세요. AWS CLI

### `list-virtual-mfa-devices`

다음 코드 예시에서는 `list-virtual-mfa-devices`을 사용하는 방법을 보여 줍니다.

AWS CLI

가상 MFA 디바이스를 나열하려면

다음 `list-virtual-mfa-devices` 명령은 현재 계정에 대해 구성된 가상 MFA 디바이스를 나열합니다.

```
aws iam list-virtual-mfa-devices
```

출력:

```
{
    "VirtualMFADevices": [
        {
            "SerialNumber": "arn:aws:iam::123456789012:mfa/ExampleMFADevice"
        },
        {
            "SerialNumber": "arn:aws:iam::123456789012:mfa/Fred"
        }
    ]
}
```

자세한 내용은 AWS IAM 사용 설명서의 가상 다중 인증(MFA) 디바이스 활성화를 참조하세요.

- 자세한 API 내용은 명령 참조ListVirtualMfaDevices의 섹션을 참조하세요. AWS CLI

### put-group-policy

다음 코드 예시에서는 put-group-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

그룹에 정책 추가

다음 put-group-policy 명령은 라는 이름의 IAM 그룹에 정책을 추가합니다Admins.

```
aws iam put-group-policy \
    --group-name Admins \
    --policy-document file://AdminPolicy.json \
    --policy-name AdminRoot
```

이 명령은 출력을 생성하지 않습니다.

정책은 AdminPolicy.json 파일의 JSON 문서로 정의됩니다. (파일 이름과 확장자는 중요하지 않습니다.)

자세한 내용은 AWS IAM 사용 설명서의 IAM 정책 관리를 참조하세요.

- 자세한 API 내용은 명령 참조PutGroupPolicy의 섹션을 참조하세요. AWS CLI

### put-role-permissions-boundary

다음 코드 예시에서는 put-role-permissions-boundary을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 사용자 지정 정책을 기반으로 권한 경계를 IAM 역할에 적용하려면

다음 put-role-permissions-boundary 예제에서는 지정된 IAM 역할에 대한 권한 경계intern-boundary로 라는 사용자 지정 정책을 적용합니다.

```
aws iam put-role-permissions-boundary \
    --permissions-boundary arn:aws:iam::123456789012:policy/intern-boundary \
    --role-name lambda-application-role
```

이 명령은 출력을 생성하지 않습니다.

예제 2: AWS 관리형 정책을 기반으로 권한 경계를 IAM 역할에 적용하려면

다음 put-role-permissions-boundary 예제에서는 AWS 관리형 PowerUserAccess 정책을 지정된 IAM 역할에 대한 권한 경계로 적용합니다.

```
aws iam put-role-permissions-boundary \
    --permissions-boundary arn:aws:iam::aws:policy/PowerUserAccess \
    --role-name x-account-admin
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 역할 수정을 참조하세요.

- 자세한 API 내용은 명령 참조PutRolePermissionsBoundary의 섹션을 참조하세요. AWS CLI

### put-role-policy

다음 코드 예시에서는 put-role-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 역할에 권한 정책을 연결하려면

다음 put-role-policy 명령은 이름이 Test-Role인 역할에 권한 정책을 추가합니다.

```
aws iam put-role-policy \
```

```
    --role-name Test-Role \
    --policy-name ExamplePolicy \
    --policy-document file://AdminPolicy.json
```

이 명령은 출력을 생성하지 않습니다.

정책은 AdminPolicy.json 파일의 JSON 문서로 정의됩니다. (파일 이름과 확장자는 중요하지 않습니다.)

신뢰 정책을 역할에 연결하려면 `update-assume-role-policy` 명령을 사용합니다.

자세한 내용은 AWS IAM 사용 설명서의 역할 수정을 참조하세요.

- 자세한 API 내용은 명령 참조PutRolePolicy의 섹션을 참조하세요. AWS CLI

### `put-user-permissions-boundary`

다음 코드 예시에서는 `put-user-permissions-boundary`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: IAM 사용자 지정 정책을 기반으로 사용자에게 권한 경계를 적용하려면

다음 `put-user-permissions-boundary` 예제에서는 지정된 IAM 사용자의 권한 경계`intern-boundary`로 라는 사용자 지정 정책을 적용합니다.

```
aws iam put-user-permissions-boundary \
    --permissions-boundary arn:aws:iam::123456789012:policy/intern-boundary \
    --user-name intern
```

이 명령은 출력을 생성하지 않습니다.

예제 2: AWS 관리형 정책을 기반으로 IAM 사용자에게 권한 경계를 적용하려면

다음 `put-user-permissions-boundary` 예제에서는 지정된 IAM 사용자의 권한 경계`PowerUserAccess`로 라는 AWS 관리형 폴리시를 적용합니다.

```
aws iam put-user-permissions-boundary \
    --permissions-boundary arn:aws:iam::aws:policy/PowerUserAccess \
    --user-name developer
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 자격 증명 권한 추가 및 제거를 참조하세요.

- 자세한 API 내용은 명령 참조PutUserPermissionsBoundary의 섹션을 참조하세요. AWS CLI

### put-user-policy

다음 코드 예시에서는 put-user-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 사용자에게 정책을 연결하려면

다음 put-user-policy 명령은 이름이 인 IAM 사용자에게 정책을 연결합니다Bob.

```
aws iam put-user-policy \
    --user-name Bob \
    --policy-name ExamplePolicy \
    --policy-document file://AdminPolicy.json
```

이 명령은 출력을 생성하지 않습니다.

정책은 AdminPolicy.json 파일의 JSON 문서로 정의됩니다. (파일 이름과 확장자는 중요하지 않습니다.)

자세한 내용은 AWS IAM 사용 설명서의 IAM 자격 증명 권한 추가 및 제거를 참조하세요.

- 자세한 API 내용은 명령 참조PutUserPolicy의 섹션을 참조하세요. AWS CLI

### remove-client-id-from-open-id-connect-provider

다음 코드 예시에서는 remove-client-id-from-open-id-connect-provider을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 IAM OpenID Connect 공급자에 IDs 등록된 클라이언트 목록에서 지정된 클라이언트 ID를 제거하려면

이 예제에서는 가 ARN인 IAM OIDC 공급자와 IDs 연결된 클라이언트 My-TestApp-3 목록에서 클라이언트 ID를 제거합니다arn: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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 OpenID Connect(OIDC) 자격 증명 공급자 생성을 참조하세요.

- 자세한 API 내용은 명령 참조RemoveClientIdFromOpenIdConnectProvider의 섹션을 참조하세요. AWS CLI

### remove-role-from-instance-profile

다음 코드 예시에서는 remove-role-from-instance-profile을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스 프로파일에서 역할 제거

다음 remove-role-from-instance-profile 명령은 이름이 ExampleInstanceProfile인 인스턴스 프로파일에서 이름이 Test-Role인 역할을 제거합니다.

```
aws iam remove-role-from-instance-profile \
    --instance-profile-name ExampleInstanceProfile \
    --role-name Test-Role
```

자세한 내용은 AWS IAM 사용 설명서의 인스턴스 프로파일 사용을 참조하세요.

- 자세한 API 내용은 명령 참조RemoveRoleFromInstanceProfile의 섹션을 참조하세요. AWS CLI

### remove-user-from-group

다음 코드 예시에서는 remove-user-from-group을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 그룹에서 사용자를 제거하려면

다음 remove-user-from-group 명령은 라는 Bob IAM 그룹에서 라는 사용자를 제거합니다Admins.

```
aws iam remove-user-from-group \
    --user-name Bob \
    --group-name Admins
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 사용자 그룹에 있는 IAM 사용자 추가 및 제거를 참조하세요.

- 자세한 API 내용은 명령 참조RemoveUserFromGroup의 섹션을 참조하세요. AWS CLI

### reset-service-specific-credential

다음 코드 예시에서는 reset-service-specific-credential을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: 요청을 하는 사용자에게 연결된 서비스별 보안 인증의 암호 재설정

다음 reset-service-specific-credential 예제에서는 요청을 수행하는 사용자에게 연결된 지정된 서비스별 자격 증명에 대해 암호학적으로 강력한 새 암호를 생성합니다.

```
aws iam reset-service-specific-credential \
    --service-specific-credential-id ACCAEXAMPLE123EXAMPLE
```

출력:

```
{
    "ServiceSpecificCredential": {
        "CreateDate": "2019-04-18T20:45:36+00:00",
        "ServiceName": "codecommit.amazonaws.com",
        "ServiceUserName": "sofia-at-123456789012",
        "ServicePassword": "+oaFsNk7tLco+C/obP9GhhcOzGcKOayTmE3LnAmAmH4=",
        "ServiceSpecificCredentialId": "ACCAEXAMPLE123EXAMPLE",
        "UserName": "sofia",
        "Status": "Active"
    }
}
```

예제 2: 지정된 사용자에게 연결된 서비스별 보안 인증의 암호 재설정

다음 reset-service-specific-credential 예제에서는 지정된 사용자에 연결된 서비스별 보안 인증에 대해 암호학적으로 강력한 새 암호를 생성합니다.

```
aws iam reset-service-specific-credential \
    --user-name sofia \
    --service-specific-credential-id ACCAEXAMPLE123EXAMPLE
```

출력:

```
{
    "ServiceSpecificCredential": {
        "CreateDate": "2019-04-18T20:45:36+00:00",
        "ServiceName": "codecommit.amazonaws.com",
        "ServiceUserName": "sofia-at-123456789012",
        "ServicePassword": "+oaFsNk7tLco+C/obP9GhhcOzGcKOayTmE3LnAmAmH4=",
        "ServiceSpecificCredentialId": "ACCAEXAMPLE123EXAMPLE",
        "UserName": "sofia",
        "Status": "Active"
    }
}
```

자세한 내용은 AWS CodeCommit 사용 설명서의 에 대한 HTTPS 연결을 위한 Git 자격 증명 생성을 CodeCommit 참조하세요.

- 자세한 API 내용은 명령 참조ResetServiceSpecificCredential의 섹션을 참조하세요. AWS CLI

### resync-mfa-device

다음 코드 예시에서는 resync-mfa-device을 사용하는 방법을 보여 줍니다.

AWS CLI

MFA 디바이스를 동기화하려면

다음 resync-mfa-device 예제에서는 IAM 사용자와 연결되어 Bob 있고 이 두 인증 코드를 제공한 인증자 프로그램과 ARN arn:aws:iam::123456789012:mfa/BobsMFADevice 연결된 MFA 디바이스를 동기화합니다.

```
aws iam resync-mfa-device \
    --user-name Bob \
    --serial-number arn:aws:iam::210987654321:mfa/BobsMFADevice \
```

```
    --authentication-code1 123456 \
    --authentication-code2 987654
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 에서 다중 인증 사용(MFA) AWS을 참조하세요.

- 자세한 API 내용은 명령 참조ResyncMfaDevice의 섹션을 참조하세요. AWS CLI

### set-default-policy-version

다음 코드 예시에서는 set-default-policy-version을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 정책의 지정된 버전을 정책의 기본 버전으로 설정

이 예제에서는 가 기본 활성 v2 버전ARNarn:aws:iam::123456789012:policy/MyPolicy인 정책의 버전을 설정합니다.

```
aws iam set-default-policy-version \
    --policy-arn arn:aws:iam::123456789012:policy/MyPolicy \
    --version-id v2
```

자세한 내용은 AWS IAM 사용 설명서의 에서 정책 및 권한을 IAM 참조하세요.

- 자세한 API 내용은 명령 참조SetDefaultPolicyVersion의 섹션을 참조하세요. AWS CLI

### set-security-token-service-preferences

다음 코드 예시에서는 set-security-token-service-preferences을 사용하는 방법을 보여 줍니다.

AWS CLI

글로벌 엔드포인트 토큰 버전을 설정하려면

다음 set-security-token-service-preferences 예제에서는 글로벌 엔드포인트STS에 대해 인증할 때 버전 2 토큰을 사용하도록 Amazon을 구성합니다.

```
aws iam set-security-token-service-preferences \
    --global-endpoint-token-version v2Token
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 AWS 리전에서 관리를 AWS STS 참조하세요.

- 자세한 API 내용은 명령 참조SetSecurityTokenServicePreferences의 섹션을 참조하세요. AWS CLI

### simulate-custom-policy

다음 코드 예시에서는 simulate-custom-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: IAM 사용자 또는 역할과 연결된 모든 IAM 정책의 효과를 시뮬레이션하려면

다음은 정책을 제공하고 변수 값을 정의하고 API 호출을 시뮬레이션하여 허용 또는 거부 여부를 확인하는 방법을 simulate-custom-policy 보여줍니다. 다음 예제에서는 지정된 날짜 및 시간 이후에만 데이터베이스 액세스를 활성화하는 정책을 보여줍니다. 시뮬레이션된 작업과 지정된 aws:CurrentTime 변수가 모두 정책의 요구 사항과 일치하므로 시뮬레이션이 성공합니다.

```
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
```

출력:

```
{
    "EvaluationResults": [
        {
            "EvalActionName": "dynamodb:CreateBackup",
            "EvalResourceName": "*",
            "EvalDecision": "allowed",
            "MatchedStatements": [
                {
                    "SourcePolicyId": "PolicyInputList.1",
                    "StartPosition": {
                        "Line": 1,
```

```
                    "Column": 38
                },
                "EndPosition": {
                    "Line": 1,
                    "Column": 167
                }
            }
        ],
        "MissingContextValues": []
    }
]
}
```

예제 2: 정책에서 금지하는 명령을 시뮬레이션하려면

다음 `simulate-custom-policy` 예제는 정책에서 금지하는 명령을 시뮬레이션한 결과를 보여줍니다. 이 예제에서는 제공된 날짜가 정책 조건에 필요한 날짜보다 앞섭니다.

```
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
```

출력:

```
{
    "EvaluationResults": [
        {
            "EvalActionName": "dynamodb:CreateBackup",
            "EvalResourceName": "*",
            "EvalDecision": "implicitDeny",
            "MatchedStatements": [],
            "MissingContextValues": []
        }
    ]
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM 정책 시뮬레이터를 사용한 IAM 정책 테스트를 참조하세요.

- 자세한 API 내용은 명령 참조SimulateCustomPolicy의 섹션을 참조하세요. AWS CLI

### simulate-principal-policy

다음 코드 예시에서는 simulate-principal-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 임의 IAM 정책의 효과를 시뮬레이션하려면

다음은 작업을 호출하고 해당 사용자와 연결된 정책이 API 작업을 허용 또는 거부할지 여부를 결정하는 사용자를 시뮬레이션하는 방법을 simulate-principal-policy 보여줍니다. 다음 예제에서는 사용자에게 codecommit:ListRepositories 작업만 허용하는 정책이 있습니다.

```
aws iam simulate-principal-policy \
    --policy-source-arn arn:aws:iam::123456789012:user/alejandro \
    --action-names codecommit:ListRepositories
```

출력:

```
{
    "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": []
        }
    ]
```

```
}
```

예제 2: 금지된 명령의 효과를 시뮬레이션하려면

다음 simulate-custom-policy 예제는 사용자 정책 중 하나에 의해 금지된 명령을 시뮬레이션한 결과를 보여줍니다. 다음 예제에서는 특정 날짜 및 시간 이후에만 DynamoDB 데이터베이스에 대한 액세스를 허용하는 정책이 사용자에게 있습니다. 시뮬레이션에는 사용자가 정책 조건에서 허용하는 값보다 빠른 aws:CurrentTime 값으로 데이터베이스에 액세스하려고 시도합니다.

```
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
```

출력:

```
{
    "EvaluationResults": [
        {
            "EvalActionName": "dynamodb:CreateBackup",
            "EvalResourceName": "*",
            "EvalDecision": "implicitDeny",
            "MatchedStatements": [],
            "MissingContextValues": []
        }
    ]
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM 정책 시뮬레이터를 사용한 IAM 정책 테스트를 참조하세요.

- 자세한 API 내용은 명령 참조SimulatePrincipalPolicy의 섹션을 참조하세요. AWS CLI

### tag-instance-profile

다음 코드 예시에서는 tag-instance-profile을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스 프로파일에 태그를 추가하려면

다음 tag-instance-profile 명령은 지정된 인스턴스 프로파일에 부서 이름이 있는 태그를 추가합니다.

```
aws iam tag-instance-profile \
    --instance-profile-name deployment-role \
    --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조TagInstanceProfile의 섹션을 참조하세요. AWS CLI

### tag-mfa-device

다음 코드 예시에서는 tag-mfa-device을 사용하는 방법을 보여 줍니다.

AWS CLI

MFA 디바이스에 태그를 추가하려면

다음 tag-mfa-device 명령은 지정된 MFA 디바이스에 부서 이름이 인 태그를 추가합니다.

```
aws iam tag-mfa-device \
    --serial-number arn:aws:iam::123456789012:mfa/alice \
    --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조TagMfaDevice의 섹션을 참조하세요. AWS CLI

### tag-open-id-connect-provider

다음 코드 예시에서는 tag-open-id-connect-provider을 사용하는 방법을 보여 줍니다.

AWS CLI

OpenID Connect(OIDC) 호환 자격 증명 공급자에 태그를 추가하려면

다음 tag-open-id-connect-provider 명령은 지정된 OIDC 자격 증명 공급자에 부서 이름이인 태그를 추가합니다.

```
aws iam tag-open-id-connect-provider \
    --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/
server.example.com \
    --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조TagOpenIdConnectProvider의 섹션을 참조하세요. AWS CLI

### tag-policy

다음 코드 예시에서는 tag-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

고객 관리형 정책에 태그를 추가하려면

다음 tag-policy 명령은 지정된 고객 관리형 정책에 부서 이름이 인 태그를 추가합니다.

```
aws iam tag-policy \
    --policy-arn arn:aws:iam::123456789012:policy/billing-access \
    --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조TagPolicy의 섹션을 참조하세요. AWS CLI

### tag-role

다음 코드 예시에서는 tag-role을 사용하는 방법을 보여 줍니다.

AWS CLI

역할에 태그 추가

다음 tag-role 명령은 지정된 역할에 Department 이름이 포함된 태그를 추가합니다.

```
aws iam tag-role --role-name my-role \
    --tags '{"Key": "Department", "Value": "Accounting"}'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조TagRole의 섹션을 참조하세요. AWS CLI

### tag-saml-provider

다음 코드 예시에서는 tag-saml-provider을 사용하는 방법을 보여 줍니다.

AWS CLI

SAML 공급자에 태그를 추가하려면

다음 tag-saml-provider 명령은 지정된 SAML 공급자에 부서 이름이 인 태그를 추가합니다.

```
aws iam tag-saml-provider \
    --saml-provider-arn arn:aws:iam::123456789012:saml-provider/ADFS \
    --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조TagSamlProvider의 섹션을 참조하세요. AWS CLI

### tag-server-certificate

다음 코드 예시에서는 tag-server-certificate을 사용하는 방법을 보여 줍니다.

AWS CLI

서버 인증서에 태그를 추가하려면

다음 tag-saml-provider 명령은 지정된 서버 인증서에 부서 이름이 있는 태그를 추가합니다.

```
aws iam tag-server-certificate \
```

```
    --server-certificate-name ExampleCertificate \
    --tags '[{"Key": "Department", "Value": "Accounting"}]'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조TagServerCertificate의 섹션을 참조하세요. AWS CLI

### tag-user

다음 코드 예시에서는 tag-user을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자에게 태그 추가

다음 tag-user 명령은 지정된 사용자에게 연관된 Department가 포함된 태그를 추가합니다.

```
aws iam tag-user \
    --user-name alice \
    --tags '{"Key": "Department", "Value": "Accounting"}'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조TagUser의 섹션을 참조하세요. AWS CLI

### untag-instance-profile

다음 코드 예시에서는 untag-instance-profile을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스 프로파일에서 태그를 제거하려면

다음 untag-instance-profile 명령은 키 이름이 'Department'인 태그를 지정된 인스턴스 프로파일에서 제거합니다.

```
aws iam untag-instance-profile \
    --instance-profile-name deployment-role \
```

```
    --tag-keys Department
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조UntagInstanceProfile의 섹션을 참조하세요. AWS CLI

### untag-mfa-device

다음 코드 예시에서는 untag-mfa-device을 사용하는 방법을 보여 줍니다.

AWS CLI

MFA 디바이스에서 태그를 제거하려면

다음 untag-mfa-device 명령은 키 이름이 'Department'인 태그를 지정된 MFA 디바이스에서 제거합니다.

```
aws iam untag-mfa-device \
    --serial-number arn:aws:iam::123456789012:mfa/alice \
    --tag-keys Department
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조UntagMfaDevice의 섹션을 참조하세요. AWS CLI

### untag-open-id-connect-provider

다음 코드 예시에서는 untag-open-id-connect-provider을 사용하는 방법을 보여 줍니다.

AWS CLI

OIDC 자격 증명 공급자에서 태그를 제거하려면

다음 untag-open-id-connect-provider 명령은 키 이름이 'Department'인 태그를 지정된 OIDC 자격 증명 공급자에서 제거합니다.

```
aws iam untag-open-id-connect-provider \
```

```
    --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/
server.example.com \
    --tag-keys Department
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조UntagOpenIdConnectProvider의 섹션을 참조하세요. AWS CLI

### untag-policy

다음 코드 예시에서는 untag-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

고객 관리형 정책에서 태그를 제거하려면

다음 untag-policy 명령은 키 이름이 'Department'인 태그를 지정된 고객 관리형 정책에서 제거합니다.

```
aws iam untag-policy \
    --policy-arn arn:aws:iam::452925170507:policy/billing-access \
    --tag-keys Department
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조UntagPolicy의 섹션을 참조하세요. AWS CLI

### untag-role

다음 코드 예시에서는 untag-role을 사용하는 방법을 보여 줍니다.

AWS CLI

역할에서 태그 제거

다음 untag-role 명령은 지정된 역할에서 키 이름이 'Department'인 모든 태그를 제거합니다.

```
aws iam untag-role \
    --role-name my-role \
```

```
    --tag-keys Department
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조UntagRole의 섹션을 참조하세요. AWS CLI

### untag-saml-provider

다음 코드 예시에서는 `untag-saml-provider`을 사용하는 방법을 보여 줍니다.

AWS CLI

SAML 공급자에서 태그를 제거하려면

다음 `untag-saml-provider` 명령은 키 이름이 'Department'인 태그를 지정된 인스턴스 프로파일에서 제거합니다.

```
aws iam untag-saml-provider \
    --saml-provider-arn arn:aws:iam::123456789012:saml-provider/ADFS \
    --tag-keys Department
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조UntagSamlProvider의 섹션을 참조하세요. AWS CLI

### untag-server-certificate

다음 코드 예시에서는 `untag-server-certificate`을 사용하는 방법을 보여 줍니다.

AWS CLI

서버 인증서에서 태그를 제거하려면

다음 `untag-server-certificate` 명령은 키 이름이 'Department'인 태그를 지정된 서버 인증서에서 제거합니다.

```
aws iam untag-server-certificate \
    --server-certificate-name ExampleCertificate \
```

```
    --tag-keys Department
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조UntagServerCertificate의 섹션을 참조하세요. AWS CLI

### untag-user

다음 코드 예시에서는 untag-user을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자에게서 태그 제거

다음 untag-user 명령은 지정된 사용자에게서 키 이름이 'Department'인 모든 태그를 제거합니다.

```
aws iam untag-user \
    --user-name alice \
    --tag-keys Department
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조UntagUser의 섹션을 참조하세요. AWS CLI

### update-access-key

다음 코드 예시에서는 update-access-key을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 사용자의 액세스 키를 활성화 또는 비활성화하려면

다음 update-access-key 명령은 이름이 인 IAM 사용자의 지정된 액세스 키(액세스 키 ID 및 보안 액세스 키)를 비활성화합니다Bob.

```
aws iam update-access-key \
    --access-key-id AKIAIOSFODNN7EXAMPLE \
```

```
    --status Inactive \
    --user-name Bob
```

이 명령은 출력을 생성하지 않습니다.

키를 비활성화하면 에 프로그래밍 방식으로 액세스하는 데 사용할 수 없습니다 AWS. 하지만 키는 계속 사용할 수 있고 다시 활성화할 수 있습니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 사용자용 액세스 키 관리를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateAccessKey의 섹션을 참조하세요. AWS CLI

### update-account-password-policy

다음 코드 예시에서는 update-account-password-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 계정 암호 정책 설정 또는 변경

다음 update-account-password-policy 명령은 최소 8자 길이를 요구하고 암호에 하나 이상의 숫자를 요구하도록 암호 정책을 설정합니다.

```
aws iam update-account-password-policy \
    --minimum-password-length 8 \
    --require-numbers
```

이 명령은 출력을 생성하지 않습니다.

계정의 암호 정책을 변경하면 계정의 IAM 사용자에 대해 생성된 새 암호에 영향을 미칩니다. 암호 정책 변경은 기존 암호에 영향을 주지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 사용자에 대한 계정 암호 정책 설정을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateAccountPasswordPolicy의 섹션을 참조하세요. AWS CLI

### update-assume-role-policy

다음 코드 예시에서는 update-assume-role-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 역할에 대한 신뢰 정책을 업데이트하려면

다음 update-assume-role-policy 명령은 Test-Role이라는 역할에 대한 신뢰 정책을 업데이트합니다.

```
aws iam update-assume-role-policy \
    --role-name Test-Role \
    --policy-document file://Test-Role-Trust-Policy.json
```

이 명령은 출력을 생성하지 않습니다.

신뢰 정책은 Test-Role-Trust-Policy.json 파일의 JSON 문서로 정의됩니다. (파일 이름과 확장자는 중요하지 않습니다.) 신뢰 정책에서 보안 주체를 지정해야 합니다.

역할에 대한 권한 정책을 업데이트하려면 put-role-policy 명령을 사용합니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 역할 생성을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateAssumeRolePolicy의 섹션을 참조하세요. AWS CLI

### update-group

다음 코드 예시에서는 update-group을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 그룹 이름을 바꾸려면

다음 update-group 명령은 IAM 그룹의 이름을 Test로 변경합니다Test-1.

```
aws iam update-group \
    --group-name Test \
    --new-group-name Test-1
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 사용자 그룹 이름 바꾸기를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateGroup의 섹션을 참조하세요. AWS CLI

### update-login-profile

다음 코드 예시에서는 update-login-profile을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 사용자의 암호를 업데이트하려면

다음 update-login-profile 명령은 이름이 인 IAM 사용자의 새 암호를 생성합니다Bob.

```
aws iam update-login-profile \
    --user-name Bob \
    --password <password>
```

이 명령은 출력을 생성하지 않습니다.

계정의 암호 정책을 설정하려면 update-account-password-policy 명령을 사용합니다. 새 암호가 계정 암호 정책을 위반하는 경우 명령은 PasswordPolicyViolation 오류를 반환합니다.

계정 암호 정책에서 허용하는 경우 IAM 사용자는 change-password 명령을 사용하여 자신의 암호를 변경할 수 있습니다.

암호를 안전한 위치에 저장합니다. 암호를 분실한 경우 복구가 불가능하며, create-login-profile 명령을 사용하여 암호를 새로 생성해야 합니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 사용자 암호 관리를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateLoginProfile의 섹션을 참조하세요. AWS CLI

### update-open-id-connect-provider-thumbprint

다음 코드 예시에서는 update-open-id-connect-provider-thumbprint을 사용하는 방법을 보여 줍니다.

AWS CLI

기존 서버 인증서 지문 목록을 새 목록으로 바꾸기

이 예제ARN에서는 새 지문을 사용할 OIDC 공급자의 인증서 지문 목록을 업데이트arn:aws:iam::123456789012:oidc-provider/example.oidcprovider.com합니다.

```
aws iam update-open-id-connect-provider-thumbprint \
    --open-id-connect-provider-arn arn:aws:iam::123456789012:oidc-provider/
example.oidcprovider.com \
```

```
    --thumbprint-list 7359755EXAMPLEabc3060bce3EXAMPLEec4542a3
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 OpenID Connect(OIDC) 자격 증명 공급자 생성을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateOpenIdConnectProviderThumbprint의 섹션을 참조하세요. AWS CLI

### update-role-description

다음 코드 예시에서는 update-role-description을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 역할의 설명을 변경하려면

다음 update-role 명령은 IAM 역할에 대한 설명을 production-role로 변경합니다Main production role.

```
aws iam update-role-description \
    --role-name production-role \
    --description 'Main production role'
```

출력:

```
{
    "Role": {
        "Path": "/",
        "RoleName": "production-role",
        "RoleId": "AROA1234567890EXAMPLE",
        "Arn": "arn:aws:iam::123456789012:role/production-role",
        "CreateDate": "2017-12-06T17:16:37+00:00",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "AWS": "arn:aws:iam::123456789012:root"
                    },
```

```
                    "Action": "sts:AssumeRole",
                    "Condition": {}
                }
            ]
        },
        "Description": "Main production role"
    }
}
```

자세한 내용은 AWS IAM 사용 설명서의 역할 수정을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateRoleDescription의 섹션을 참조하세요. AWS CLI

### update-role

다음 코드 예시에서는 update-role을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 역할의 설명 또는 세션 기간을 변경하려면

다음 update-role 명령은 IAM 역할에 대한 설명을 production-role로 변경Main production role하고 최대 세션 기간을 12시간으로 설정합니다.

```
aws iam update-role \
    --role-name production-role \
    --description 'Main production role' \
    --max-session-duration 43200
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 역할 수정을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateRole의 섹션을 참조하세요. AWS CLI

### update-saml-provider

다음 코드 예시에서는 update-saml-provider을 사용하는 방법을 보여 줍니다.

AWS CLI

기존 SAML 공급자의 메타데이터 문서를 업데이트하려면

이 예제ARN에서는 가 IAM 인 SAML 공급자를 파일 의 새 SAML 메타데이터 문서arn:aws:iam::123456789012:saml-provider/SAMLADFS로 업데이트합니다SAMLMetaData.xml.

```
aws iam update-saml-provider \
    --saml-metadata-document file://SAMLMetaData.xml \
    --saml-provider-arn arn:aws:iam::123456789012:saml-provider/SAMLADFS
```

출력:

```
{
    "SAMLProviderArn": "arn:aws:iam::123456789012:saml-provider/SAMLADFS"
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM SAML 자격 증명 공급자 생성을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateSamlProvider의 섹션을 참조하세요. AWS CLI

### update-server-certificate

다음 코드 예시에서는 update-server-certificate을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 계정에서 서버 인증서의 경로 또는 이름을 변경하려면

다음 update-server-certificate 명령은 인증서의 이름을 mServerCertificate에서 myUpdatedServerCertificate로 변경합니다. 또한 Amazon CloudFront 서비스에서 액세스할 수 /cloudfront/ 있도록 경로를 로 변경합니다. 이 명령은 출력을 생성하지 않습니다. list-server-certificates 명령을 실행하여 업데이트 결과를 볼 수 있습니다.

```
aws-iam update-server-certificate \
    --server-certificate-name myServerCertificate \
    --new-server-certificate-name myUpdatedServerCertificate \
    --new-path /cloudfront/
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 에서 서버 인증서 관리를 IAM 참조하세요.

- 자세한 API 내용은 명령 참조UpdateServerCertificate의 섹션을 참조하세요. AWS CLI

### update-service-specific-credential

다음 코드 예시에서는 update-service-specific-credential을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 요청 사용자의 서비스별 자격 증명 상태를 업데이트하려면

다음 update-service-specific-credential 예제에서는 에 요청하는 사용자의 지정된 자격 증명 상태를 변경합니다Inactive.

```
aws iam update-service-specific-credential \
    --service-specific-credential-id ACCAEXAMPLE123EXAMPLE \
    --status Inactive
```

이 명령은 출력을 생성하지 않습니다.

예제 2: 지정된 사용자의 서비스별 자격 증명의 상태를 업데이트하려면

다음 update-service-specific-credential 예제에서는 지정된 사용자의 자격 증명 상태를 비활성으로 변경합니다.

```
aws iam update-service-specific-credential \
    --user-name sofia \
    --service-specific-credential-id ACCAEXAMPLE123EXAMPLE \
    --status Inactive
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS CodeCommit 사용 설명서의 에 대한 HTTPS 연결을 위한 Git 자격 증명 생성을 CodeCommit 참조하세요.

- 자세한 API 내용은 명령 참조UpdateServiceSpecificCredential의 섹션을 참조하세요. AWS CLI

### update-signing-certificate

다음 코드 예시에서는 update-signing-certificate을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 사용자의 서명 인증서를 활성화 또는 비활성화하려면

다음 update-signing-certificate 명령은 라는 IAM 사용자에 대해 지정된 서명 인증서를 비활성화합니다Bob.

```
aws iam update-signing-certificate \
    --certificate-id TA7SMP42TDN5Z26OBPJE7EXAMPLE \
    --status Inactive \
    --user-name Bob
```

서명 인증서의 ID를 가져오려면 list-signing-certificates 명령을 사용합니다.

자세한 내용은 Amazon EC2 사용 설명서의 서명 인증서 관리를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateSigningCertificate의 섹션을 참조하세요. AWS CLI

### update-ssh-public-key

다음 코드 예시에서는 update-ssh-public-key을 사용하는 방법을 보여 줍니다.

AWS CLI

SSH 퍼블릭 키의 상태를 변경하려면

다음 update-ssh-public-key 명령은 지정된 퍼블릭 키의 상태를 로 변경합니다Inactive.

```
aws iam update-ssh-public-key \
    --user-name sofia \
    --ssh-public-key-id APKA1234567890EXAMPLE \
    --status Inactive
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 및 SSH와 함께 SSH 키 사용을 CodeCommit 참조하세요.

- 자세한 API 내용은 명령 참조UpdateSshPublicKey의 섹션을 참조하세요. AWS CLI

### update-user

다음 코드 예시에서는 update-user을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 사용자 이름을 변경하려면

다음 update-user 명령은 IAM 사용자의 이름을 Bob로 변경합니다Robert.

```
aws iam update-user \
    --user-name Bob \
    --new-user-name Robert
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 사용자 그룹 이름 바꾸기를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateUser의 섹션을 참조하세요. AWS CLI

### upload-server-certificate

다음 코드 예시에서는 upload-server-certificate을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 계정에 서버 인증서를 업로드하려면

다음 upload-server-certificate 명령은 서버 인증서를 AWS 계정에 업로드합니다. 이 예제에서 인증서는 public_key_cert_file.pem 파일에 있고, 연결된 프라이빗 키가 my_private_key.pem 파일에 있으며, CA(인증 기관)에서 제공하는 인증서 체인은 my_certificate_chain_file.pem 파일에 있습니다. 파일 업로드가 완료되면 라는 이름으로 파일을 사용할 수 있습니다myServerCertificate. file://로 시작하는 파라미터는 명령에 파일 내용을 읽고 해당 내용을 파일 이름 대신 파라미터 값으로 사용하도록 지시합니다.

```
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
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 사용 IAM 설명서의 서버 인증서 생성, 업로드 및 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조UploadServerCertificate의 섹션을 참조하세요. AWS CLI

### upload-signing-certificate

다음 코드 예시에서는 upload-signing-certificate을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 사용자의 서명 인증서를 업로드하려면

다음 upload-signing-certificate 명령은 이름이 인 IAM 사용자의 서명 인증서를 업로드합니다Bob.

```
aws iam upload-signing-certificate \
    --user-name Bob \
    --certificate-body file://certificate.pem
```

출력:

```
{
    "Certificate": {
        "UserName": "Bob",
        "Status": "Active",
        "CertificateBody": "-----BEGIN CERTIFICATE-----<certificate-body>-----END
 CERTIFICATE-----",
        "CertificateId": "TA7SMP42TDN5Z26OBPJE7EXAMPLE",
        "UploadDate": "2013-06-06T21:40:08.121Z"
    }
}
```

인증서는 certificate.pem이라는 이름의 파일에 PEM 형식이 있습니다.

자세한 내용은 사용 IAM 설명서의 사용자 서명 인증서 생성 및 업로드를 참조하세요.

- 자세한 API 내용은 명령 참조UploadSigningCertificate의 섹션을 참조하세요. AWS CLI

### upload-ssh-public-key

다음 코드 예시에서는 upload-ssh-public-key을 사용하는 방법을 보여 줍니다.

AWS CLI

SSH 퍼블릭 키를 업로드하고 사용자와 연결하려면

다음 upload-ssh-public-key 명령은 파일에 있는 퍼블릭 키를 업로드sshkey.pub하여 사용자에게 연결합니다sofia.

```
aws iam upload-ssh-public-key \
    --user-name sofia \
    --ssh-public-key-body file://sshkey.pub
```

출력:

```
{
    "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"
    }
}
```

이 명령에 적합한 형식으로 키를 생성하는 방법에 대한 자세한 내용은 AWS CodeCommit 사용 설명서의 SSH 및 Linux, macOS 또는 Unix: Git 및 또는 Windows에 대한 퍼블릭 및 프라이빗 키 설정 CodeCommitSSH: Git 및 에 대한 퍼블릭 및 프라이빗 키 설정을 CodeCommit 참조하세요.

- 자세한 API 내용은 명령 참조UploadSshPublicKey의 섹션을 참조하세요. AWS CLI

## IAM 를 사용하여 분석기 예제 액세스 AWS CLI

다음 코드 예제에서는 IAM Access Analyzer AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `apply-archive-rule`

다음 코드 예시에서는 `apply-archive-rule`을 사용하는 방법을 보여 줍니다.

AWS CLI

아카이브 규칙 기준을 충족하는 기존 결과에 아카이브 규칙을 적용하려면

다음 `apply-archive-rule` 예제에서는 아카이브 규칙 기준을 충족하는 기존 결과에 아카이브 규칙을 적용합니다.

```
aws accessanalyzer apply-archive-rule \
    --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
UnusedAccess-ConsoleAnalyzer-organization \
    --rule-name MyArchiveRule
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 아카이브 규칙을 참조하세요.

- 자세한 API 내용은 명령 참조ApplyArchiveRule의 섹션을 참조하세요. AWS CLI

### `cancel-policy-generation`

다음 코드 예시에서는 `cancel-policy-generation`을 사용하는 방법을 보여 줍니다.

AWS CLI

요청된 정책 생성을 취소하려면

다음 cancel-policy-generation 예제에서는 요청된 정책 생성 작업 ID를 취소합니다.

```
aws accessanalyzer cancel-policy-generation \
    --job-id 923a56b0-ebb8-4e80-8a3c-a11ccfbcd6f2
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM Access Analyzer 정책 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CancelPolicyGeneration의 섹션을 참조하세요. AWS CLI

### check-access-not-granted

다음 코드 예시에서는 check-access-not-granted을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 액세스가 정책에서 허용되지 않는지 확인하려면

다음 check-access-not-granted 예제에서는 지정된 액세스가 정책에서 허용되지 않는지 확인합니다.

```
aws accessanalyzer check-access-not-granted \
    --policy-document file://myfile.json \
    --access actions="s3:DeleteBucket","s3:GetBucketLocation" \
    --policy-type IDENTITY_POLICY
```

myfile.json의 콘텐츠:

```
{
    "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/*"
            ]
```

```
        }
    ]
}
```

출력:

```
{
    "result": "PASS",
    "message": "The policy document does not grant access to perform one or more of
 the listed actions."
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM Access Analyzer를 사용한 액세스 미리 보기를 APIs 참조하세요.

- 자세한 API 내용은 명령 참조CheckAccessNotGranted의 섹션을 참조하세요. AWS CLI

### check-no-new-access

다음 코드 예시에서는 check-no-new-access을 사용하는 방법을 보여 줍니다.

AWS CLI

기존 정책과 비교할 때 업데이트된 정책에 대해 새 액세스가 허용되는지 확인하려면

다음 check-no-new-access 예제에서는 기존 정책과 비교할 때 업데이트된 정책에 대해 새 액세스가 허용되는지 여부를 확인합니다.

```
aws accessanalyzer check-no-new-access \
    --existing-policy-document file://existing-policy.json \
    --new-policy-document file://new-policy.json \
    --policy-type IDENTITY_POLICY
```

existing-policy.json의 콘텐츠:

```
{
    "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/*"
            ]
        }
    ]
}
```

new-policy.json의 콘텐츠:

```
{
    "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/*"
            ]
        }
    ]
}
```

출력:

```
{
    "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
        }
    ]
```

```
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM Access Analyzer를 사용한 액세스 미리 보기를 APIs 참조하세요.

- 자세한 API 내용은 명령 참조CheckNoNewAccess의 섹션을 참조하세요. AWS CLI

### check-no-public-access

다음 코드 예시에서는 check-no-public-access을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 정책이 지정된 리소스 유형에 대한 퍼블릭 액세스 권한을 부여할 수 있는지 확인하려면

다음 check-no-public-access 예제에서는 리소스 정책이 지정된 리소스 유형에 대한 퍼블릭 액세스 권한을 부여할 수 있는지 확인합니다.

```
aws accessanalyzer check-no-public-access \
    --policy-document file://check-no-public-access-myfile.json \
    --resource-type AWS::S3::Bucket
```

myfile.json의 콘텐츠:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CheckNoPublicAccess",
            "Effect": "Allow",
            "Principal": { "AWS": "arn:aws:iam::111122223333:user/JohnDoe" },
            "Action": [
                "s3:GetObject"
            ]
        }
    ]
}
```

출력:

```
{
```

```
    "result": "PASS",
    "message": "The resource policy does not grant public access for the given
 resource type."
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM Access Analyzer를 사용한 액세스 미리 보기를 APIs 참조하세요.

- 자세한 API 내용은 명령 참조CheckNoPublicAccess의 섹션을 참조하세요. AWS CLI

### `create-access-preview`

다음 코드 예시에서는 `create-access-preview`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 권한을 배포하기 전에 리소스에 대한 IAM Access Analyzer 결과를 미리 볼 수 있는 액세스 미리 보기를 생성하려면

다음 `create-access-preview` 예제에서는 AWS 계정에 리소스 권한을 배포하기 전에 리소스에 대한 IAM Access Analyzer 결과를 미리 볼 수 있는 액세스 미리 보기를 생성합니다.

```
aws accessanalyzer create-access-preview \
    --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account \
    --configurations file://myfile.json
```

`myfile.json`의 콘텐츠:

```
{
    "arn:aws:s3:::DOC-EXAMPLE-BUCKET": {
        "s3Bucket": {
            "bucketPolicy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\":\"Allow\",\"Principal\":{\"AWS\":[\"arn:aws:iam::111122223333:root\"]},\"Action
\":[\"s3:PutObject\",\"s3:PutObjectAcl\"],\"Resource\":\"arn:aws:s3:::DOC-EXAMPLE-
BUCKET/*\"}]}",
            "bucketPublicAccessBlock": {
                "ignorePublicAcls": true,
                "restrictPublicBuckets": true
            },
            "bucketAclGrants": [
```

```
                {
                    "grantee": {
                        "id":
 "79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be"
                    },
                    "permission": "READ"
                }
            ]
        }
    }
}
```

출력:

```
{
    "id": "3c65eb13-6ef9-4629-8919-a32043619e6b"
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM Access Analyzer를 사용한 액세스 미리 보기를 APIs 참조하세요.

- 자세한 API 내용은 명령 참조CreateAccessPreview의 섹션을 참조하세요. AWS CLI

### `create-analyzer`

다음 코드 예시에서는 `create-analyzer`을 사용하는 방법을 보여 줍니다.

AWS CLI

분석기를 생성하려면

다음 `create-analyzer` 예제에서는 AWS 계정에 분석기를 생성합니다.

```
aws accessanalyzer create-analyzer \
    --analyzer-name example \
    --type ACCOUNT
```

출력:

```
{
    "arn": "arn:aws:access-analyzer:us-east-2:111122223333:analyzer/example"
```

```
}
```

자세한 내용은 AWS IAM 사용 설명서의 AWS Identity and Access Management Access Analyzer 결과 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조CreateAnalyzer의 섹션을 참조하세요. AWS CLI

### create-archive-rule

다음 코드 예시에서는 create-archive-rule을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 분석기에 대한 아카이브 규칙을 생성하려면

다음 create-archive-rule 예제에서는 AWS 계정에서 지정된 분석기에 대한 아카이브 규칙을 생성합니다.

```
aws accessanalyzer create-archive-rule \
    --analyzer-name UnusedAccess-ConsoleAnalyzer-organization \
    --rule-name MyRule \
    --filter '{"resource": {"contains": ["Cognito"]}, "resourceType": {"eq":
 ["AWS::IAM::Role"]}}'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 아카이브 규칙을 참조하세요.

- 자세한 API 내용은 명령 참조CreateArchiveRule의 섹션을 참조하세요. AWS CLI

### delete-analyzer

다음 코드 예시에서는 delete-analyzer을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 분석기를 삭제하려면

다음 delete-analyzer 예제에서는 AWS 계정에서 지정된 분석기를 삭제합니다.

```
aws accessanalyzer delete-analyzer \
    --analyzer-name example
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 아카이브 규칙을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteAnalyzer의 섹션을 참조하세요. AWS CLI

### delete-archive-rule

다음 코드 예시에서는 delete-archive-rule을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 아카이브 규칙을 삭제하려면

다음 delete-archive-rule 예제에서는 AWS 계정에서 지정된 아카이브 규칙을 삭제합니다.

```
aws accessanalyzer delete-archive-rule \
    --analyzer-name UnusedAccess-ConsoleAnalyzer-organization \
    --rule-name MyRule
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 아카이브 규칙을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteArchiveRule의 섹션을 참조하세요. AWS CLI

### get-access-preview

다음 코드 예시에서는 get-access-preview을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 분석기의 액세스 미리 보기에 대한 정보를 검색하려면

다음 get-access-preview 예제에서는 AWS 계정에서 지정된 분석기의 액세스 미리 보기에 대한 정보를 검색합니다.

```
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
```

출력:

```
{
    "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\":
[{\"Effect\":\"Allow\",\"Principal\":{\"AWS\":[\"arn:aws:iam::111122223333:root\"]},
\"Action\":[\"s3:PutObject\",\"s3:PutObjectAcl\"],\"Resource\":\"arn:aws:s3:::DOC-
EXAMPLE-BUCKET/*\"}]}",
                    "bucketAclGrants": [
                        {
                            "permission": "READ",
                            "grantee": {
                                "id":
 "79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be"
                            }
                        }
                    ],
                    "bucketPublicAccessBlock": {
                        "ignorePublicAcls": true,
                        "restrictPublicBuckets": true
                    }
                }
            }
        },
        "createdAt": "2024-02-17T00:18:44+00:00",
        "status": "COMPLETED"
    }
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM Access Analyzer를 사용한 액세스 미리 보기를 APIs 참조하세요.

- 자세한 API 내용은 명령 참조GetAccessPreview의 섹션을 참조하세요. AWS CLI

### `get-analyzed-resource`

다음 코드 예시에서는 `get-analyzed-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

분석된 리소스에 대한 정보를 검색하려면

다음 `get-analyzed-resource` 예제에서는 AWS 계정에서 분석된 리소스에 대한 정보를 검색합니다.

```
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
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS IAM 사용 설명서의 AWS Identity and Access Management Access Analyzer 사용을 참조하세요.

- 자세한 API 내용은 명령 참조GetAnalyzedResource의 섹션을 참조하세요. AWS CLI

### `get-analyzer`

다음 코드 예시에서는 `get-analyzer`을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 분석기에 대한 정보를 검색하려면

다음 `get-analyzer` 예제에서는 AWS 계정에서 지정된 분석기에 대한 정보를 검색합니다.

```
aws accessanalyzer get-analyzer \
    --analyzer-name ConsoleAnalyzer-account
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS IAM 사용 설명서의 AWS Identity and Access Management Access Analyzer 사용을 참조하세요.

- 자세한 API 내용은 명령 참조GetAnalyzer의 섹션을 참조하세요. AWS CLI

### get-archive-rule

다음 코드 예시에서는 get-archive-rule을 사용하는 방법을 보여 줍니다.

AWS CLI

아카이브 규칙에 대한 정보를 검색하려면

다음 get-archive-rule 예제에서는 AWS 계정의 아카이브 규칙에 대한 정보를 검색합니다.

```
aws accessanalyzer get-archive-rule \
    --analyzer-name UnusedAccess-ConsoleAnalyzer-organization \
    --rule-name MyArchiveRule
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS IAM 사용 설명서의 아카이브 규칙을 참조하세요.

- 자세한 API 내용은 명령 참조GetArchiveRule의 섹션을 참조하세요. AWS CLI

### get-finding-v2

다음 코드 예시에서는 get-finding-v2을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 결과에 대한 정보를 검색하려면

다음 get-finding-v2 예제는 AWS 계정에서 지정된 결과에 대한 정보를 에트리브합니다.

```
aws accessanalyzer get-finding-v2 \
    --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-organization \
    --id 0910eedb-381e-4e95-adda-0d25c19e6e90
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS IAM 사용 설명서의 결과 검토를 참조하세요.

- API 자세한 내용은 AWS CLI 명령 참조의 GetFindingV2를 참조하세요.

### get-finding

다음 코드 예시에서는 get-finding을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 결과에 대한 정보를 검색하려면

다음 get-finding 예제는 AWS 계정에서 지정된 결과에 대한 정보를 에트리브합니다.

```
aws accessanalyzer get-finding \
    --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-organization \
    --id 0910eedb-381e-4e95-adda-0d25c19e6e90
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS IAM 사용 설명서의 결과 검토를 참조하세요.

- 자세한 API 내용은 명령 참조GetFinding의 섹션을 참조하세요. AWS CLI

### get-generated-policy

다음 코드 예시에서는 get-generated-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

`StartPolicyGeneration`를 사용하여 생성된 정책을 검색하려면 API

다음 get-generated-policy 예제에서는 AWS 계정에서 를 사용하여 StartPolicyGeneration API 생성된 정책을 검색합니다.

```
aws accessanalyzer get-generated-policy \
    --job-id c557dc4a-0338-4489-95dd-739014860ff9
```

출력:


```
{
    "generatedPolicyResult": {
        "generatedPolicies": [
            {
                "policy": "{\"Version\":\"2012-10-17\",\"Statement\":
[{\"Sid\":\"SupportedServiceSid0\",\"Effect\":\"Allow\",\"Action\":
[\"access-analyzer:GetAnalyzer\",\"access-analyzer:ListAnalyzers\",
\"access-analyzer:ListArchiveRules\",\"access-analyzer:ListFindings
\",\"cloudtrail:DescribeTrails\",\"cloudtrail:GetEventDataStore\",
\"cloudtrail:GetEventSelectors\",\"cloudtrail:GetInsightSelectors
\",\"cloudtrail:GetTrailStatus\",\"cloudtrail:ListChannels\",
\"cloudtrail:ListEventDataStores\",\"cloudtrail:ListQueries\",\"cloudtrail:ListTags
\",\"cloudtrail:LookupEvents\",\"ec2:DescribeRegions\",\"iam:GetAccountSummary
\",\"iam:GetOpenIDConnectProvider\",\"iam:GetRole\",\"iam:ListAccessKeys\",
\"iam:ListAccountAliases\",\"iam:ListOpenIDConnectProviders\",\"iam:ListRoles
\",\"iam:ListSAMLProviders\",\"kms:ListAliases\",\"s3:GetBucketLocation\",
\"s3:ListAllMyBuckets\"],\"Resource\":\"*\"}]}"
            }
        ],
        "properties": {
            "cloudTrailProperties": {
                "endTime": "2024-02-14T22:44:40+00:00",
                "startTime": "2024-02-13T00:30:00+00:00",
                "trailProperties": [
                    {
                        "allRegions": true,
                        "cloudTrailArn": "arn:aws:cloudtrail:us-
west-2:111122223333:trail/my-trail",
                        "regions": []
                    }
                ]
            },
            "isComplete": false,
            "principalArn": "arn:aws:iam::111122223333:role/Admin"
        }
    },
    "jobDetails": {
        "completedOn": "2024-02-14T22:47:01+00:00",
        "jobId": "c557dc4a-0338-4489-95dd-739014860ff9",
        "startedOn": "2024-02-14T22:44:41+00:00",
        "status": "SUCCEEDED"
    }
```

```
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM Access Analyzer 정책 생성을 참조하세요.

- 자세한 API 내용은 명령 참조GetGeneratedPolicy의 섹션을 참조하세요. AWS CLI

### `list-access-preview-findings`

다음 코드 예시에서는 `list-access-preview-findings`을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 액세스 미리 보기에서 생성된 액세스 미리 보기 조사 결과 목록을 검색하려면

다음 `list-access-preview-findings` 예제에서는 AWS 계정의 지정된 액세스 미리 보기에서 생성된 액세스 미리 보기 조사 결과 목록을 검색합니다.

```
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
```

출력:

```
{
    "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"
                }
            ]
        }
    ]
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM Access Analyzer를 사용한 액세스 미리 보기를 APIs 참조하세요.

- 자세한 API 내용은 명령 참조ListAccessPreviewFindings의 섹션을 참조하세요. AWS CLI

### `list-access-previews`

다음 코드 예시에서는 `list-access-previews`을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 분석기에 대한 액세스 미리 보기 목록을 검색하려면

다음 `list-access-previews` 예제에서는 AWS 계정에서 지정된 분석기에 대한 액세스 미리 보기 목록을 검색합니다.

```
aws accessanalyzer list-access-previews \
    --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM Access Analyzer를 사용한 액세스 미리 보기를 APIs 참조하세요.

- 자세한 API 내용은 명령 참조ListAccessPreviews의 섹션을 참조하세요. AWS CLI

### `list-analyzed-resources`

다음 코드 예시에서는 `list-analyzed-resources`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 위젯을 나열하려면

다음 `list-analyzed-resources` 예제에서는 AWS 계정에서 사용 가능한 위젯을 나열합니다.

```
aws accessanalyzer list-analyzed-resources \
    --analyzer-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account \
    --resource-type AWS::IAM::Role
```

출력:

```
{
    "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"
```

```
        }
    ]
}
```

자세한 내용은 AWS IAM 사용 설명서의 AWS Identity and Access Management Access Analyzer 사용을 참조하세요.

- 자세한 API 내용은 명령 참조ListAnalyzedResources의 섹션을 참조하세요. AWS CLI

### `list-analyzers`

다음 코드 예시에서는 `list-analyzers`을 사용하는 방법을 보여 줍니다.

AWS CLI

분석기 목록을 검색하려면

다음 `list-analyzers` 예제에서는 AWS 계정의 분석기 목록을 검색합니다.

```
aws accessanalyzer list-analyzers
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS IAM 사용 설명서의 AWS Identity and Access Management Access Analyzer 사용을 참조하세요.

- 자세한 API 내용은 명령 참조ListAnalyzers의 섹션을 참조하세요. AWS CLI

### `list-archive-rules`

다음 코드 예시에서는 `list-archive-rules`을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 분석기에 대해 생성된 아카이브 규칙 목록을 검색하려면

다음 `list-archive-rules` 예제에서는 AWS 계정에서 지정된 분석기에 대해 생성된 아카이브 규칙 목록을 검색합니다.

```
aws accessanalyzer list-archive-rules \
    --analyzer-name UnusedAccess-ConsoleAnalyzer-organization
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS IAM 사용 설명서의 AWS Identity and Access Management Access Analyzer 사용을 참조하세요.

- 자세한 API 내용은 명령 참조ListArchiveRules의 섹션을 참조하세요. AWS CLI

### `list-findings-v2`

다음 코드 예시에서는 `list-findings-v2`을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 분석기에서 생성된 결과 목록을 검색하려면

다음 `list-findings-v2` 예제에서는 AWS 계정의 지정된 분석기에서 생성된 조사 결과 목록을 검색합니다. 이 예제에서는 이름이 를 포함하는 IAM 역할만 포함하도록 결과를 필터링합니다Cognito.

```
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"]}}'
```

출력:

```
{
    "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"
        }
    ]
```

```
}
```

자세한 내용은 AWS IAM 사용 설명서의 AWS Identity and Access Management Access Analyzer 사용을 참조하세요.

- API 자세한 내용은 AWS CLI 명령 참조의 ListFindingsV2를 참조하세요.

### `list-findings`

다음 코드 예시에서는 `list-findings`을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 분석기에서 생성된 결과 목록을 검색하려면

다음 `list-findings` 예제에서는 AWS 계정의 지정된 분석기에서 생성된 조사 결과 목록을 검색합니다. 이 예제에서는 이름이 를 포함하는 IAM 역할만 포함하도록 결과를 필터링합니다Cognito.

```
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"]}}'
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS IAM 사용 설명서의 AWS Identity and Access Management Access Analyzer 사용을 참조하세요.

- 자세한 API 내용은 명령 참조ListFindings의 섹션을 참조하세요. AWS CLI

### `list-policy-generations`

다음 코드 예시에서는 `list-policy-generations`을 사용하는 방법을 보여 줍니다.

AWS CLI

지난 7일 동안 요청된 모든 정책 생성을 나열하려면

다음 `list-policy-generations` 예제에서는 AWS 계정에서 지난 7일 동안 요청된 모든 정책 생성을 나열합니다.

```
aws accessanalyzer list-policy-generations
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM 액세스 분석기 정책 생성을 참조하세요.

- 자세한 API 내용은 명령 참조ListPolicyGenerations의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 리소스에 적용된 태그 목록을 검색하려면

다음 `list-tags-for-resource` 예제에서는 AWS 계정의 지정된 리소스에 적용된 태그 목록을 검색합니다.

```
aws accessanalyzer list-tags-for-resource \
```

```
    --resource-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account
```

출력:

```
{
    "tags": {
        "Zone-of-trust": "Account",
        "Name": "ConsoleAnalyzer"
    }
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM 액세스 분석기 정책 생성을 참조하세요.

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### start-policy-generation

다음 코드 예시에서는 start-policy-generation을 사용하는 방법을 보여 줍니다.

AWS CLI

정책 생성 요청을 시작하려면

다음 start-policy-generation 예제에서는 AWS 계정에서 정책 생성 요청을 시작합니다.

```
aws accessanalyzer start-policy-generation \
    --policy-generation-details '{"principalArn":"arn:aws:iam::111122223333:role/
Admin"}' \
    --cloud-trail-details file://myfile.json
```

myfile.json의 콘텐츠:

```
{
    "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"
```

```
        }
    ]
}
```

출력:

```
{
    "jobId": "c557dc4a-0338-4489-95dd-739014860ff9"
}
```

자세한 내용은 AWS IAM 사용 설명서의 IAM 액세스 분석기 정책 생성을 참조하세요.

- 자세한 API 내용은 명령 참조StartPolicyGeneration의 섹션을 참조하세요. AWS CLI

### `start-resource-scan`

다음 코드 예시에서는 `start-resource-scan`을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 리소스에 적용된 정책 스캔을 즉시 시작하려면

다음 `start-resource-scan` 예제에서는 AWS 계정의 지정된 리소스에 적용된 정책의 스캔을 순식간에 시작합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 IAM 액세스 분석기 정책 생성을 참조하세요.

- 자세한 API 내용은 명령 참조StartResourceScan의 섹션을 참조하세요. AWS CLI

### `tag-resource`

다음 코드 예시에서는 `tag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 리소스에 태그를 추가하려면

다음 tag-resource 예제에서는 AWS 계정의 지정된 리소스에 태그를 추가합니다.

```
aws accessanalyzer tag-resource \
    --resource-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account \
    --tags Environment=dev,Purpose=testing
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 AWS Identity and Access Management Access Analyzer 사용을 참조하세요.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### untag-resource

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 리소스에서 태그를 제거하려면

다음 untag-resource 예제에서는 AWS 계정의 지정된 리소스에서 태그를 제거합니다.

```
aws accessanalyzer untag-resource \
    --resource-arn arn:aws:access-analyzer:us-west-2:111122223333:analyzer/
ConsoleAnalyzer-account \
    --tag-keys Environment Purpose
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 AWS Identity and Access Management Access Analyzer 사용을 참조하세요.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-archive-rule

다음 코드 예시에서는 update-archive-rule을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 아카이브 규칙의 기준 및 값을 업데이트하려면

다음 update-archive-rule 예제에서는 AWS 계정에서 지정된 아카이브 규칙의 기준과 값을 업데이트합니다.

```
aws accessanalyzer update-archive-rule \
    --analyzer-name UnusedAccess-ConsoleAnalyzer-organization \
    --rule-name MyArchiveRule \
    --filter '{"resource": {"contains": ["Cognito"]}, "resourceType": {"eq":
 ["AWS::IAM::Role"]}}'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 아카이브 규칙을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateArchiveRule의 섹션을 참조하세요. AWS CLI

### update-findings

다음 코드 예시에서는 update-findings을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 조사 결과의 상태를 업데이트하려면

다음 update-findings 예제에서는 AWS 계정에서 지정된 조사 결과의 상태를 업데이트합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IAM 사용 설명서의 AWS Identity and Access Management Access Analyzer 사용을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateFindings의 섹션을 참조하세요. AWS CLI

### validate-policy

다음 코드 예시에서는 validate-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

정책 검증을 요청하고 조사 결과 목록을 반환하려면

다음 `validate-policy` 예제에서는 정책의 검증을 요청하고 조사 결과 목록을 반환합니다. 예제의 정책은 웹 ID 페더레이션에 사용되는 Amazon Cognito 역할에 대한 역할 신뢰 정책입니다. 신뢰 정책에서 생성된 조사 결과는 잘못된 수임 역할 작업이 사용되어 빈 `Sid` 요소 값과 정책 보안 주체가 일치하지 않는 와 관련이 있습니다`sts:AssumeRole`. Cognito와 함께 사용할 올바른 역할 가정 작업은 입니다`sts:AssumeRoleWithWebIdentity`.

```
aws accessanalyzer validate-policy \
    --policy-document file://myfile.json \
    --policy-type RESOURCE_POLICY
```

`myfile.json`의 콘텐츠:

```
{
    "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"
                }
            }
        }
    ]
}
```

출력:

```
{
    "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
                        }
                    }
                }
```

```
            ]
        }
    ]
}
```

자세한 내용은 AWS IAM 사용 설명서의 정책 검증 확인을 참조하세요.

- 자세한 API 내용은 명령 참조ValidatePolicy의 섹션을 참조하세요. AWS CLI

## 를 사용한 Image Builder 예제 AWS CLI

다음 코드 예제에서는 Image Builder AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `create-component`

다음 코드 예시에서는 `create-component`을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 요소를 생성하려면

다음 `create-component` 예제에서는 JSON 문서 파일을 사용하는 구성 요소를 생성하고 Amazon S3 버킷에 업로드되는 YAML 형식의 구성 요소 문서를 참조합니다.

```
aws imagebuilder create-component \
    --cli-input-json file://create-component.json
```

`create-component.json`의 콘텐츠:

```
{
    "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"
}
```

출력:

```
{
    "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"
}
```

자세한 내용은 EC2 Image Builder 사용 설명서의 를 사용하여 Image Builder 이미지 파이프라인 설정 및 관리를 AWS CLI 참조하세요. EC2

- 자세한 API 내용은 명령 참조CreateComponent의 섹션을 참조하세요. AWS CLI

### `create-distribution-configuration`

다음 코드 예시에서는 `create-distribution-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

배포 구성을 생성하려면

다음 `create-distribution-configuration` 예제에서는 JSON 파일을 사용하여 배포 구성을 생성합니다.

```
aws imagebuilder create-distribution-configuration \
    --cli-input-json file:/create-distribution-configuration.json
```

`create-distribution-configuration.json`의 콘텐츠:

```
{
    "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"
                    ]
                }
            }
        }
    ]
}
```

출력:

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "distributionConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:distribution-configuration/myexampledistribution"
}
```

자세한 내용은 EC2 Image Builder 사용 설명서의 를 사용하여 Image Builder 이미지 파이프라인 설정 및 관리를 AWS CLI 참조하세요. EC2

- 자세한 API 내용은 명령 참조CreateDistributionConfiguration의 섹션을 참조하세요. AWS CLI

### `create-image-pipeline`

다음 코드 예시에서는 `create-image-pipeline`을 사용하는 방법을 보여 줍니다.

AWS CLI

이미지 파이프라인을 생성하려면

다음 `create-image-pipeline` 예제에서는 JSON 파일을 사용하여 이미지 파이프라인을 생성합니다.

```
aws imagebuilder create-image-pipeline \
    --cli-input-json file://create-image-pipeline.json
```

`create-image-pipeline.json`의 콘텐츠:

```
{
    "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"
}
```

출력:

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "imagePipelineArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline"
}
```

자세한 내용은 EC2 Image Builder 사용 설명서의 를 사용하여 Image Builder 이미지 파이프라인 설정 및 관리를 AWS CLI 참조하세요. EC2

- 자세한 API 내용은 명령 참조CreateImagePipeline의 섹션을 참조하세요. AWS CLI

### create-image-recipe

다음 코드 예시에서는 create-image-recipe을 사용하는 방법을 보여 줍니다.

AWS CLI

레시피를 생성하려면

다음 create-image-recipe 예제에서는 JSON 파일을 사용하여 이미지 레시피를 생성합니다. 구성 요소는 지정된 순서대로 설치됩니다.

```
aws imagebuilder create-image-recipe \
    --cli-input-json file://create-image-recipe.json
```

create-image-recipe.json의 콘텐츠:

```
{
    "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"
}
```

출력:

```
{
    "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"
}
```

자세한 내용은 EC2 Image Builder 사용 설명서의 를 사용하여 Image Builder 이미지 파이프라인 설정 및 관리를 AWS CLI 참조하세요. EC2

- 자세한 API 내용은 명령 참조CreateImageRecipe의 섹션을 참조하세요. AWS CLI

### `create-image`

다음 코드 예시에서는 `create-image`을 사용하는 방법을 보여 줍니다.

AWS CLI

이미지를 생성하려면

다음 `create-image` 예제에서는 이미지를 생성합니다.

```
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
```

출력:

```
{
    "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"
}
```

자세한 내용은 EC2 Image Builder 사용 설명서의 를 사용하여 Image Builder 이미지 파이프라인 설정 및 관리를 AWS CLI 참조하세요. EC2

- 자세한 API 내용은 명령 참조CreateImage의 섹션을 참조하세요. AWS CLI

### create-infrastructure-configuration

다음 코드 예시에서는 create-infrastructure-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

인프라 구성을 생성하려면

다음 create-infrastructure-configuration 예제에서는 JSON 파일을 사용하여 인프라 구성을 생성합니다.

```
aws imagebuilder create-infrastructure-configuration \
    --cli-input-json file://create-infrastructure-configuration.json
```

create-infrastructure-configuration.json의 콘텐츠:

```
{
    "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"
}
```

출력:

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "clientToken": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:infrastructure-configuration/myexampleinfrastructure"
}
```

자세한 내용은 EC2 Image Builder 사용 설명서의 를 사용하여 Image Builder 이미지 파이프라인 설정 및 관리를 AWS CLI 참조하세요. EC2

- 자세한 API 내용은 명령 참조CreateInfrastructureConfiguration의 섹션을 참조하세요. AWS CLI

### `delete-component`

다음 코드 예시에서는 `delete-component`을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 요소를 삭제하려면

다음 `delete-component` 예제에서는 를 지정하여 구성 요소 빌드 버전을 삭제합니다ARN.

```
aws imagebuilder delete-component \
    --component-build-version-arn arn:aws:imagebuilder:us-
west-2:123456789012:component/myexamplecomponent/2019.12.02/1
```

출력:

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "componentBuildVersionArn": "arn:aws:imagebuilder:us-
west-2:123456789012:component/myexamplecomponent/2019.12.02/1"
}
```

자세한 내용은 EC2 Image Builder 사용 설명서의 를 사용하여 Image Builder 이미지 파이프라인 설정 및 관리를 AWS CLI 참조하세요. EC2

- 자세한 API 내용은 명령 참조DeleteComponent의 섹션을 참조하세요. AWS CLI

### delete-image-pipeline

다음 코드 예시에서는 delete-image-pipeline을 사용하는 방법을 보여 줍니다.

AWS CLI

이미지 파이프라인을 삭제하려면

다음 delete-image-pipeline 예제에서는 를 지정하여 이미지 파이프라인을 삭제합니다ARN.

```
aws imagebuilder delete-image-pipeline \
    --image-pipeline-arn arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
my-example-pipeline
```

출력:

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "imagePipelineArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline"
}
```

자세한 내용은 EC2 Image Builder 사용 설명서의 를 사용하여 Image Builder 이미지 파이프라인 설정 및 관리를 AWS CLI 참조하세요. EC2

- 자세한 API 내용은 명령 참조DeleteImagePipeline의 섹션을 참조하세요. AWS CLI

### delete-image-recipe

다음 코드 예시에서는 delete-image-recipe을 사용하는 방법을 보여 줍니다.

AWS CLI

이미지 레시피를 삭제하려면

다음 delete-image-recipe 예제에서는 를 지정하여 이미지 레시피를 삭제합니다ARN.

```
aws imagebuilder delete-image-recipe \
    --image-recipe-arn arn:aws:imagebuilder:us-east-1:123456789012:image-recipe/
mybasicrecipe/2019.12.03
```

출력:

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "imageRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/
mybasicrecipe/2019.12.03"
}
```

자세한 내용은 EC2 Image Builder 사용 설명서의 를 사용하여 Image Builder 이미지 파이프라인 설정 및 관리를 AWS CLI 참조하세요. EC2

- 자세한 API 내용은 명령 참조DeleteImageRecipe의 섹션을 참조하세요. AWS CLI

### delete-image

다음 코드 예시에서는 delete-image을 사용하는 방법을 보여 줍니다.

AWS CLI

이미지를 삭제하려면

다음 delete-image 예제에서는 를 지정하여 이미지 빌드 버전을 삭제합니다ARN.

```
aws imagebuilder delete-image \
    --image-build-version-arn arn:aws:imagebuilder:us-west-2:123456789012:image/my-
example-image/2019.12.02/1
```

출력:

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "imageBuildVersionArn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/1"
}
```

자세한 내용은 EC2 Image Builder 사용 설명서의 를 사용하여 Image Builder 이미지 파이프라인 설정 및 관리를 AWS CLI 참조하세요. EC2

- 자세한 API 내용은 명령 참조DeleteImage의 섹션을 참조하세요. AWS CLI

### `delete-infrastructure-configuration`

다음 코드 예시에서는 `delete-infrastructure-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

인프라 구성을 삭제하려면

다음 `delete-infrastructure-configuration` 예제에서는 를 지정하여 이미지 파이프라인을 삭제합니다ARN.

```
aws imagebuilder delete-infrastructure-configuration \
    --infrastructure-configuration-arn arn:aws:imagebuilder:us-
east-1:123456789012:infrastructure-configuration/myexampleinfrastructure
```

출력:

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "infrastructureConfigurationArn": "arn:aws:imagebuilder:us-
west-2:123456789012:infrastructure-configuration/myexampleinfrastructure"
}
```

자세한 내용은 EC2 Image Builder 사용 설명서의 를 사용하여 Image Builder 이미지 파이프라인 설정 및 관리를 AWS CLI 참조하세요. EC2

- 자세한 API 내용은 명령 참조DeleteInfrastructureConfiguration의 섹션을 참조하세요. AWS CLI

### `get-component-policy`

다음 코드 예시에서는 `get-component-policy`을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 요소 정책 세부 정보를 가져오려면

다음 `get-component-policy` 예제에서는 를 지정하여 구성 요소 정책의 세부 정보를 나열합니다ARN.

```
aws imagebuilder get-component-policy \
    --component-arn arn:aws:imagebuilder:us-west-2:123456789012:component/my-example-component/2019.12.03/1
```

출력:

```
{
    "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" ] } ] }"
}
```

자세한 내용은 EC2 Image Builder 사용 설명서의 <https://docs.aws.amazon.com/imagebuilder/latest/userguide/managing-image-builder-cli.html>`__를 AWS CLI 사용하여 Image Builder 이미지 파이프라인 설정 및 관리를 참조하세요. EC2

- 자세한 API 내용은 명령 참조GetComponentPolicy의 섹션을 참조하세요. AWS CLI

### get-component

다음 코드 예시에서는 get-component을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 요소 세부 정보를 가져오려면

다음 get-component 예제에서는 를 지정하여 구성 요소의 세부 정보를 나열합니다ARN.

```
aws imagebuilder get-component \
    --component-build-version-arn arn:aws:imagebuilder:us-west-2:123456789012:component/component-name/1.0.0/1
```

출력:

```
{
    "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": {}
    }
}
```

자세한 내용은 EC2 Image Builder 사용 설명서의 를 사용하여 Image Builder 이미지 파이프라인 설정 및 관리를 AWS CLI 참조하세요. EC2

- 자세한 API 내용은 명령 참조GetComponent의 섹션을 참조하세요. AWS CLI

### get-distribution-configuration

다음 코드 예시에서는 get-distribution-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

배포 구성의 세부 정보를 가져오려면

다음 get-distribution-configuration 예제에서는 배포 구성을 지정하여 배포 구성의 세부 정보를 표시합니다ARN.

```
aws imagebuilder get-distribution-configuration \
    --distribution-configuration-arn arn:aws:imagebuilder:us-
west-2:123456789012:distribution-configuration/myexampledistribution
```

출력:

```
{
    "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": {}
    }
}
```

자세한 내용은 EC2 Image Builder 사용 설명서의 를 사용하여 Image Builder 이미지 파이프라인 설정 및 관리를 AWS CLI 참조하세요. EC2

- 자세한 API 내용은 명령 참조GetDistributionConfiguration의 섹션을 참조하세요. AWS CLI

### get-image-pipeline

다음 코드 예시에서는 get-image-pipeline을 사용하는 방법을 보여 줍니다.

AWS CLI

이미지 파이프라인 세부 정보를 가져오려면

다음 get-image-pipeline 예제에서는 를 지정하여 이미지 파이프라인의 세부 정보를 나열합니다ARN.

```
aws imagebuilder get-image-pipeline \
    --image-pipeline-arn arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline
```

출력:

```
{
    "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": {}
    }
}
```

자세한 내용은 EC2 Image Builder 사용 설명서의 를 사용하여 Image Builder 이미지 파이프라인 설정 및 관리를 AWS CLI 참조하세요. EC2

- 자세한 API 내용은 명령 참조GetImagePipeline의 섹션을 참조하세요. AWS CLI

### get-image-policy

다음 코드 예시에서는 get-image-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

이미지 정책 세부 정보를 가져오려면

다음 get-image-policy 예제에서는 를 지정하여 이미지 정책의 세부 정보를 나열합니다ARN.

```
aws imagebuilder get-image-policy \
    --image-arn arn:aws:imagebuilder:us-west-2:123456789012:image/my-example-image/2019.12.03/1
```

출력:

```
{
    "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" ] } ] }"
}
```

자세한 내용은 EC2 Image Builder 사용 설명서의 를 사용하여 Image Builder 이미지 파이프라인 설정 및 관리를 AWS CLI 참조하세요. EC2

- 자세한 API 내용은 명령 참조GetImagePolicy의 섹션을 참조하세요. AWS CLI

### get-image-recipe-policy

다음 코드 예시에서는 get-image-recipe-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

이미지 레시피 정책 세부 정보를 가져오려면

다음 `get-image-recipe-policy` 예제에서는 를 지정하여 이미지 레시피 정책의 세부 정보를 나열합니다ARN.

```
aws imagebuilder get-image-recipe-policy \
    --image-recipe-arn arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/my-example-image-recipe/2019.12.03/1
```

출력:

```
{
    "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" ] } ] }"
}
```

자세한 내용은 EC2 Image Builder 사용 설명서의 를 사용하여 Image Builder 이미지 파이프라인 설정 및 관리를 AWS CLI 참조하세요. EC2

- 자세한 API 내용은 명령 참조GetImageRecipePolicy의 섹션을 참조하세요. AWS CLI

### `get-image`

다음 코드 예시에서는 `get-image`을 사용하는 방법을 보여 줍니다.

AWS CLI

이미지 세부 정보를 가져오려면

다음 `get-image` 예제에서는 를 지정하여 이미지의 세부 정보를 나열합니다ARN.

```
aws imagebuilder get-image \
    --image-build-version-arn arn:aws:imagebuilder:us-west-2:123456789012:image/mybasicrecipe/2019.12.03/1
```

출력:

```
{
    "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": {}
    }
}
```

자세한 내용은 EC2 Image Builder 사용 설명서의 를 사용하여 Image Builder 이미지 파이프라인 설정 및 관리를 AWS CLI 참조하세요. EC2

- 자세한 API 내용은 명령 참조GetImage의 섹션을 참조하세요. AWS CLI

### `get-infrastructure-configuration`

다음 코드 예시에서는 `get-infrastructure-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

인프라 구성 세부 정보를 가져오려면

다음 get-infrastructure-configuration 예제에서는 를 지정하여 인프라 구성의 세부 정보를 나열합니다ARN.

```
aws imagebuilder get-infrastructure-configuration \
    --infrastructure-configuration-arn arn:aws:imagebuilder:us-
west-2:123456789012:infrastructure-configuration/myexampleinfrastructure
```

출력:

```
{
    "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": {}
    }
}
```

자세한 내용은 EC2 Image Builder 사용 설명서의 를 사용하여 Image Builder 이미지 파이프라인 설정 및 관리를 AWS CLI 참조하세요. EC2

- 자세한 API 내용은 명령 참조GetInfrastructureConfiguration의 섹션을 참조하세요. AWS CLI

**import-component**

다음 코드 예시에서는 import-component을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 요소를 가져오려면

다음 import-component 예제에서는 JSON 파일을 사용하여 기존 스크립트를 가져옵니다.

```
aws imagebuilder import-component \
    --cli-input-json file://import-component.json
```

import-component.json의 콘텐츠:

```
{
    "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"
}
```

출력:

```
{
    "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"
}
```

자세한 내용은 EC2 Image Builder 사용 설명서의 를 사용하여 Image Builder 이미지 파이프라인 설정 및 관리를 AWS CLI 참조하세요. EC2

- 자세한 API 내용은 명령 참조ImportComponent의 섹션을 참조하세요. AWS CLI

### `list-component-build-versions`

다음 코드 예시에서는 `list-component-build-versions`을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 요소 빌드 버전을 나열하려면

다음 `list-component-build-versions` 예제에서는 특정 의미 버전이 있는 구성 요소 빌드 버전을 나열합니다.

```
aws imagebuilder list-component-build-versions --component-
version-arn arn:aws:imagebuilder:us-west-2:123456789012:component/
myexamplecomponent/2019.12.02
```

출력:

```
{
    "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"
            }
        }
    ]
}
```

자세한 내용은 EC2 Image Builder 사용 설명서의 를 사용하여 Image Builder 이미지 파이프라인 설정 및 관리를 AWS CLI 참조하세요. EC2

- 자세한 API 내용은 명령 참조ListComponentBuildVersions의 섹션을 참조하세요. AWS CLI

### list-components

다음 코드 예시에서는 list-components을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 구성 요소 시맨틱 버전을 나열하려면

다음 list-components 예제에서는 액세스 권한이 있는 모든 구성 요소 의미 체계 버전을 나열합니다. 선택적으로 사용자가 소유한 구성 요소, Amazon이 소유한 구성 요소 또는 다른 계정에서 공유한 구성 요소를 나열할지 여부를 기준으로 필터링할 수 있습니다.

```
aws imagebuilder list-components
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 EC2 Image Builder 사용 설명서의 를 사용하여 Image Builder 이미지 파이프라인 설정 및 관리를 AWS CLI 참조하세요. EC2

- 자세한 API 내용은 명령 참조ListComponents의 섹션을 참조하세요. AWS CLI

### list-distribution-configurations

다음 코드 예시에서는 list-distribution-configurations을 사용하는 방법을 보여 줍니다.

AWS CLI

배포를 나열하려면

다음 `list-distribution-configurations` 예제에서는 모든 배포를 나열합니다.

```
aws imagebuilder list-distribution-configurations
```

출력:

```
{
    "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"
            }
        }
    ]
}
```

자세한 내용은 EC2 Image Builder 사용 설명서의 를 사용하여 Image Builder 이미지 파이프라인 설정 및 관리를 AWS CLI 참조하세요. EC2

- 자세한 API 내용은 명령 참조ListDistributionConfigurations의 섹션을 참조하세요. AWS CLI

## `list-image-build-versions`

다음 코드 예시에서는 `list-image-build-versions`을 사용하는 방법을 보여 줍니다.

AWS CLI

이미지 빌드 버전을 나열하려면

다음 `list-image-build-versions` 예제에서는 의미 버전이 있는 모든 이미지 빌드 버전을 나열합니다.

```
aws imagebuilder list-image-build-versions \
    --image-version-arn arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03
```

출력:

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "imageSummaryList": [
        {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/7",
            "name": "MyBasicRecipe",
            "version": "2019.12.03/7",
            "platform": "Windows",
            "state": {
                "status": "FAILED",
                "reason": "Can't start SSM Automation for arn
 arn:aws:imagebuilder:us-west-2:123456789012:image/mybasicrecipe/2019.12.03/7 during
 building. Parameter \"iamInstanceProfileName\" has a null value."
            },
            "owner": "123456789012",
            "dateCreated": "2020-02-19T18:56:11.511Z",
            "outputResources": {
                "amis": []
            },
            "tags": {}
        },
        {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/6",
            "name": "MyBasicRecipe",
            "version": "2019.12.03/6",
            "platform": "Windows",
            "state": {
                "status": "FAILED",
                "reason": "An internal error has occurred."
            },
            "owner": "123456789012",
            "dateCreated": "2020-02-18T22:49:08.142Z",
            "outputResources": {
                "amis": [
                    {
```

```
                        "region": "us-west-2",
                        "image": "ami-a1b2c3d4567890ab",
                        "name": "MyBasicRecipe 2020-02-18T22-49-38.704Z",
                        "description": "This example image recipe creates a Windows
 2016 image."
                    },
                    {
                        "region": "us-west-2",
                        "image": "ami-a1b2c3d4567890ab",
                        "name": "Name 2020-02-18T22-49-08.131Z",
                        "description": "Copies AMI to eu-west-2 and exports to S3"
                    },
                    {
                        "region": "eu-west-2",
                        "image": "ami-a1b2c3d4567890ab",
                        "name": "My 6 image 2020-02-18T22-49-08.131Z",
                        "description": "Copies AMI to eu-west-2 and exports to S3"
                    }
                ]
            },
            "tags": {}
        },
        {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/5",
            "name": "MyBasicRecipe",
            "version": "2019.12.03/5",
            "platform": "Windows",
            "state": {
                "status": "AVAILABLE"
            },
            "owner": "123456789012",
            "dateCreated": "2020-02-18T16:51:48.403Z",
            "outputResources": {
                "amis": [
                    {
                        "region": "us-west-2",
                        "image": "ami-a1b2c3d4567890ab",
                        "name": "MyBasicRecipe 2020-02-18T16-52-18.965Z",
                        "description": "This example image recipe creates a Windows
 2016 image."
                    }
                ]
            },
```

```
            "tags": {}
        },
        {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/4",
            "name": "MyBasicRecipe",
            "version": "2019.12.03/4",
            "platform": "Windows",
            "state": {
                "status": "AVAILABLE"
            },
            "owner": "123456789012",
            "dateCreated": "2020-02-18T16:50:01.827Z",
            "outputResources": {
                "amis": [
                    {
                        "region": "us-west-2",
                        "image": "ami-a1b2c3d4567890ab",
                        "name": "MyBasicRecipe 2020-02-18T16-50-32.280Z",
                        "description": "This example image recipe creates a Windows
 2016 image."
                    }
                ]
            },
            "tags": {}
        },
        {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/3",
            "name": "MyBasicRecipe",
            "version": "2019.12.03/3",
            "platform": "Windows",
            "state": {
                "status": "AVAILABLE"
            },
            "owner": "123456789012",
            "dateCreated": "2020-02-14T23:14:13.597Z",
            "outputResources": {
                "amis": [
                    {
                        "region": "us-west-2",
                        "image": "ami-a1b2c3d4567890ab",
                        "name": "MyBasicRecipe 2020-02-14T23-14-44.243Z",
```

```
                        "description": "This example image recipe creates a Windows
 2016 image."
                    }
                ]
            },
            "tags": {}
        },
        {
            "arn": "arn:aws:imagebuilder:us-west-2:123456789012:image/
mybasicrecipe/2019.12.03/2",
            "name": "MyBasicRecipe",
            "version": "2019.12.03/2",
            "platform": "Windows",
            "state": {
                "status": "FAILED",
                "reason": "SSM execution 'a1b2c3d4-5678-90ab-cdef-EXAMPLE11111'
 failed with status = 'Failed' and failure message = 'Step fails when it is
 verifying the command has completed. Command a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
 returns unexpected invocation result: \n{Status=[Failed], ResponseCode=[1],
 Output=[\n----------ERROR-------\nfailed to run commands: exit status 1],
 OutputPayload=[{\"Status\":\"Failed\",\"ResponseCode\":1,\"Output\":\"\
\n----------ERROR-------\\nfailed to run commands: exit status 1\",\"CommandId\":
\"a1b2c3d4-5678-90ab-cdef-EXAMPLE11111\"}], CommandId=[a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111]}. Please refer to Automation Service Troubleshooting Guide for more
 diagnosis details.'"
            },
            "owner": "123456789012",
            "dateCreated": "2020-02-14T22:57:42.593Z",
            "outputResources": {
                "amis": []
            },
            "tags": {}
        }
    ]
}
```

자세한 내용은 EC2 Image Builder 사용 설명서의 를 사용하여 Image Builder 이미지 파이프라인 설정 및 관리를 AWS CLI 참조하세요. EC2

- 자세한 API 내용은 명령 참조ListImageBuildVersions의 섹션을 참조하세요. AWS CLI

### list-image-pipeline-images

다음 코드 예시에서는 list-image-pipeline-images을 사용하는 방법을 보여 줍니다.

AWS CLI

이미지 파이프라인 이미지를 나열하려면

다음 list-image-pipeline-images 예제에서는 특정 이미지 파이프라인에서 생성한 모든 이미지를 나열합니다.

```
aws imagebuilder list-image-pipeline-images \
    --image-pipeline-arn arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline
```

출력:

```
{
    "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"
            }
        }
    ]
}
```

자세한 내용은 EC2 Image Builder 사용 설명서의 를 사용하여 Image Builder 이미지 파이프라인 설정 및 관리를 AWS CLI 참조하세요. EC2

- 자세한 API 내용은 명령 참조ListImagePipelineImages의 섹션을 참조하세요. AWS CLI

### `list-image-recipes`

다음 코드 예시에서는 `list-image-recipes`을 사용하는 방법을 보여 줍니다.

AWS CLI

이미지 레시피를 나열하려면

다음 `list-image-recipes` 예제에서는 모든 이미지 레시피를 나열합니다.

```
aws imagebuilder list-image-recipes
```

출력:

```
{
    "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"
            }
        }
    ]
}
```

자세한 내용은 EC2 Image Builder 사용 설명서의 를 사용하여 Image Builder 이미지 파이프라인 설정 및 관리를 AWS CLI 참조하세요. EC2

- 자세한 API 내용은 명령 참조ListImageRecipes의 섹션을 참조하세요. AWS CLI

### `list-images`

다음 코드 예시에서는 `list-images`을 사용하는 방법을 보여 줍니다.

AWS CLI

이미지를 나열하려면

다음 `list-images` 예제에서는 액세스할 수 있는 모든 의미 버전을 나열합니다.

```
aws imagebuilder list-images
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 EC2 Image Builder 사용 설명서의 를 사용하여 Image Builder 이미지 파이프라인 설정 및 관리를 AWS CLI 참조하세요. EC2

- 자세한 API 내용은 명령 참조ListImages의 섹션을 참조하세요. AWS CLI

### `list-infrastructure-configurations`

다음 코드 예시에서는 `list-infrastructure-configurations`을 사용하는 방법을 보여 줍니다.

AWS CLI

인프라 구성을 나열하려면

다음 `list-infrastructure-configurations` 예제에서는 모든 인프라 구성을 나열합니다.

```
aws imagebuilder list-infrastructure-configurations
```

출력:

```
{
    "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"
            }
        }
    ]
}
```

자세한 내용은 EC2 Image Builder 사용 설명서의 를 사용하여 Image Builder 이미지 파이프라인 설정 및 관리를 AWS CLI 참조하세요. EC2

- 자세한 API 내용은 명령 참조ListInfrastructureConfigurations의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

특정 리소스에 대한 태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 특정 리소스에 대한 모든 태그를 나열합니다.

```
aws imagebuilder list-tags-for-resource \
    --resource-arn arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline
```

출력:

```
{
    "tags": {
        "KeyName": "KeyValue"
    }
}
```

자세한 내용은 EC2 Image Builder 사용 설명서의 를 사용하여 Image Builder 이미지 파이프라인 설정 및 관리를 AWS CLI 참조하세요. EC2

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### put-component-policy

다음 코드 예시에서는 put-component-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 요소에 리소스 정책을 적용하려면

다음 put-component-policy 명령은 빌드 구성 요소의 교차 계정 공유를 활성화하기 위해 빌드 구성 요소에 리소스 정책을 적용합니다. RAM CLI 명령을 사용하는 것이 좋습니다create-resource-share. EC2 Image Builder CLI 명령 을 사용하는 경우 RAM CLI promote-resource-share-create-from-policy 리소스가 공유되는 모든 보안 주체에게 리소스를 표시하려면 명령도 사용해야 put-component-policy합니다.

```
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" ] } ] }'
```

출력:

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "componentArn": "arn:aws:imagebuilder:us-west-2:123456789012:component/
examplecomponent/2019.12.02/1"
}
```

자세한 내용은 EC2 Image Builder 사용 설명서의 를 사용하여 Image Builder 이미지 파이프라인 설정 및 관리를 AWS CLI 참조하세요. EC2

- 자세한 API 내용은 명령 참조PutComponentPolicy의 섹션을 참조하세요. AWS CLI

### put-image-policy

다음 코드 예시에서는 put-image-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

이미지에 리소스 정책을 적용하려면

다음 put-image-policy 명령은 이미지에 리소스 정책을 적용하여 이미지의 교차 계정 공유를 활성화합니다. RAM CLI 명령을 사용하는 것이 좋습니다 create-resource-share. EC2 Image Builder CLI 명령 을 사용하는 경우 리소스가 공유되는 모든 보안 주체에게 리소스를 표시하려면 정책 promote-resource-share-create에서 RAM CLI 명령을 사용해야 put-image-policy합니다.

```
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" ] } ] }'
```

출력:

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "imageArn": "arn:aws:imagebuilder:us-west-2:123456789012:image/example-
image/2019.12.02/1"
}
```

자세한 내용은 EC2 Image Builder 사용 설명서의 를 사용하여 Image Builder 이미지 파이프라인 설정 및 관리를 AWS CLI 참조하세요. EC2

- 자세한 API 내용은 명령 참조PutImagePolicy의 섹션을 참조하세요. AWS CLI

### put-image-recipe-policy

다음 코드 예시에서는 put-image-recipe-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

이미지 레시피에 리소스 정책을 적용하려면

다음 put-image-recipe-policy 명령은 이미지 레시피에 리소스 정책을 적용하여 이미지 레시피의 교차 계정 공유를 활성화합니다. RAM CLI 명령을 사용하는 것이 좋습니다create-resource-share. EC2 Image Builder CLI 명령 을 사용하는 경우 promote-resource-share-create-from-policy 리소스가 공유되는 모든 보안 주체에게 리소스를 표시하려면 RAM CLI 명령도 사용해야 put-image-recipe-policy합니다.

```
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" ] } ] }'
```

출력:

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "imageRecipeArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-recipe/
example-image-recipe/2019.12.02/1"
}
```

자세한 내용은 EC2 Image Builder 사용 설명서의 를 사용하여 Image Builder 이미지 파이프라인 설정 및 관리를 AWS CLI 참조하세요. EC2

- 자세한 API 내용은 명령 참조PutImageRecipePolicy의 섹션을 참조하세요. AWS CLI

### start-image-pipeline-execution

다음 코드 예시에서는 start-image-pipeline-execution을 사용하는 방법을 보여 줍니다.

AWS CLI

이미지 파이프라인을 수동으로 시작하려면

다음 start-image-pipeline-execution 예제에서는 이미지 파이프라인을 수동으로 시작합니다.

```
aws imagebuilder start-image-pipeline-execution \
    --image-pipeline-arn arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline
```

출력:

```
{
    "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"
}
```

자세한 내용은 EC2 Image Builder 사용 설명서의 를 사용하여 Image Builder 이미지 파이프라인 설정 및 관리를 AWS CLI 참조하세요. EC2

- 자세한 API 내용은 명령 참조StartImagePipelineExecution의 섹션을 참조하세요. AWS CLI

### tag-resource

다음 코드 예시에서는 tag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 태그를 지정하려면

다음 tag-resource 예제에서는 JSON 파일을 사용하여 EC2 Image Builder에 리소스를 추가하고 태그를 지정합니다.

```
aws imagebuilder tag-resource \
```

```
    --cli-input-json file://tag-resource.json
```

tag-resource.json의 콘텐츠:

```
{
    "resourceArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline",
    "tags": {
        "KeyName: "KeyValue"
    }
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 EC2 Image Builder 사용 설명서의 를 사용하여 Image Builder 이미지 파이프라인 설정 및 관리를 AWS CLI 참조하세요. EC2

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### untag-resource

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에서 태그를 제거하려면

다음 untag-resource 예제에서는 JSON 파일을 사용하여 리소스에서 태그를 제거합니다.

```
aws imagebuilder untag-resource \
    --cli-input-json file://tag-resource.json
```

untag-resource.json의 콘텐츠:

```
{
    "resourceArn": "arn:aws:imagebuilder:us-west-2:123456789012:image-pipeline/
mywindows2016pipeline",
    "tagKeys": [
        "KeyName"
    ]
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 EC2 Image Builder 사용 설명서의 를 사용하여 Image Builder 이미지 파이프라인 설정 및 관리를 AWS CLI 참조하세요. EC2

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-distribution-configuration

다음 코드 예시에서는 update-distribution-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

배포 구성을 업데이트하려면

다음 update-distribution-configuration 예제에서는 JSON 파일을 사용하여 배포 구성을 업데이트합니다.

```
aws imagebuilder update-distribution-configuration \
    --cli-input-json file://update-distribution-configuration.json
```

update-distribution-configuration.json의 콘텐츠:

```
{
    "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}}"
            }
        }
    ]
```

```
}
```

출력:

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

자세한 내용은 EC2 Image Builder 사용 설명서의 를 사용하여 Image Builder 이미지 파이프라인 설정 및 관리를 AWS CLI 참조하세요. EC2

- 자세한 API 내용은 명령 참조UpdateDistributionConfiguration의 섹션을 참조하세요. AWS CLI

### update-image-pipeline

다음 코드 예시에서는 update-image-pipeline을 사용하는 방법을 보여 줍니다.

AWS CLI

이미지 파이프라인을 업데이트하려면

다음 update-image-pipeline 예제에서는 JSON 파일을 사용하여 이미지 파이프라인을 업데이트합니다.

```
aws imagebuilder update-image-pipeline \
    --cli-input-json file://update-image-pipeline.json
```

update-image-pipeline.json의 콘텐츠:

```
{
    "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"
}
```

출력:

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

자세한 내용은 EC2 Image Builder 사용 설명서의 를 사용하여 Image Builder 이미지 파이프라인 설정 및 관리를 AWS CLI 참조하세요. EC2

- 자세한 API 내용은 명령 참조UpdateImagePipeline의 섹션을 참조하세요. AWS CLI

### update-infrastructure-configuration

다음 코드 예시에서는 update-infrastructure-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

인프라 구성을 업데이트하려면

다음 update-infrastructure-configuration 예제에서는 JSON 파일을 사용하여 인프라 구성을 업데이트합니다.

```
aws imagebuilder update-infrastructure-configuration \
    --cli-input-json file:/update-infrastructure-configuration.json
```

update-infrastructure-configuration.json의 콘텐츠:

```
{
    "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"
}
```

출력:

```
{
    "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

자세한 내용은 EC2 Image Builder 사용 설명서의 를 사용하여 Image Builder 이미지 파이프라인 설정 및 관리를 AWS CLI 참조하세요. EC2

- 자세한 API 내용은 명령 참조UpdateInfrastructureConfiguration의 섹션을 참조하세요. AWS CLI

## 를 사용한 Incident Manager 예제 AWS CLI

다음 코드 예제에서는 Incident Manager와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### `create-replication-set`

다음 코드 예시에서는 `create-replication-set`을 사용하는 방법을 보여 줍니다.

AWS CLI

복제 세트를 생성하려면

다음 `create-replication-set` 예제에서는 Incident Manager가 Amazon Web Services 계정의 데이터를 복제하고 암호화하는 데 사용하는 복제 세트를 생성합니다. 이 예제에서는 복제 세트를 생성하는 동안 us-east-1 및 us-east-2 리전을 사용합니다.

```
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"}}'
```

출력:

```
{
    "replicationSetArns": [
        "arn:aws:ssm-incidents::111122223333:replication-set/c4bcb603-4bf9-
bb3f-413c-08df53673b57"
    ]
}
```

자세한 내용은 Incident Manager 사용 설명서의 Incident Manager 복제 세트 사용을 참조하세요.

- 자세한 API 내용은 명령 참조CreateReplicationSet의 섹션을 참조하세요. AWS CLI

### `create-response-plan`

다음 코드 예시에서는 `create-response-plan`을 사용하는 방법을 보여 줍니다.

AWS CLI

대응 계획을 생성하려면

다음 `create-response-plan` 예제에서는 지정된 세부 정보가 포함된 응답 계획을 생성합니다.

```
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"]'
```

출력:

```
{
    "arn": "arn:aws:ssm-incidents::111122223333:response-plan/example-response"
}
```

자세한 내용은 Incident Manager 사용 설명서의 인시던트 준비를 참조하세요.

- 자세한 API 내용은 명령 참조CreateResponsePlan의 섹션을 참조하세요. AWS CLI

### `create-timeline-event`

다음 코드 예시에서는 `create-timeline-event`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 사용자 지정 타임라인 이벤트 생성

다음 `create-timeline-event` 예제에서는 지정된 인시던트에 대해 지정된 시간에 사용자 지정 타임라인 이벤트를 생성합니다.

```
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"
```

출력:

```
{
    "eventId": "c0bcc885-a41d-eb01-b4ab-9d2deEXAMPLE",
    "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-record/
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4EXAMPLE"
}
```

예제 2: 인시던트 메모를 사용하여 타임라인 이벤트를 생성하려면

다음 create-timeline-event 예제에서는 '인시던트 노트' 패널에 나열된 타임라인 이벤트를 생성합니다.

```
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"}]'
```

출력:

```
{
    "eventId": "a41dc885-c0bc-b4ab-eb01-de9d2EXAMPLE",
    "incidentRecordArn": "arn:aws:ssm-incidents::111122223333:incident-record/
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4EXAMPLE"
}
```

자세한 내용은 Incident Manager 사용 설명서의 인시던트 세부 정보를 참조하세요.

- 자세한 API 내용은 명령 참조CreateTimelineEvent의 섹션을 참조하세요. AWS CLI

### delete-incident-record

다음 코드 예시에서는 delete-incident-record을 사용하는 방법을 보여 줍니다.

AWS CLI

인시던트 레코드를 삭제하려면

다음 delete-incident-record 예제에서는 지정된 인시던트 레코드를 삭제합니다.

```
aws ssm-incidents delete-incident-record \
    --arn "arn:aws:ssm-incidents::111122223333:incident-record/Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Incident Manager 사용 설명서의 인시던트 추적을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteIncidentRecord의 섹션을 참조하세요. AWS CLI

### `delete-replication-set`

다음 코드 예시에서는 `delete-replication-set`을 사용하는 방법을 보여 줍니다.

AWS CLI

복제 세트를 삭제하려면

다음 `delete-replication-set` 예제에서는 Amazon Web Services 계정에서 복제 세트를 삭제합니다. 복제 세트를 삭제하면 모든 Incident Manager 데이터도 삭제됩니다. 실행 취소할 수 없습니다.

```
aws ssm-incidents delete-replication-set \
    --arn "arn:aws:ssm-incidents::111122223333:replication-set/c4bcb603-4bf9-bb3f-413c-08df53673b57"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Incident Manager 사용 설명서의 Incident Manager 복제 세트 사용을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteReplicationSet의 섹션을 참조하세요. AWS CLI

### `delete-resource-policy`

다음 코드 예시에서는 `delete-resource-policy`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 정책을 삭제하려면

다음 `delete-resource-policy` 예제에서는 응답 계획에서 리소스 정책을 삭제합니다. 이렇게 하면 대응 계획이 공유된 보안 주체 또는 조직의 액세스가 취소됩니다.

```
aws ssm-incidents delete-resource-policy \
    --policy-id "be8b57191f0371f1c6827341aa3f0a03" \
    --resource-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-
Response-Plan"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Incident Manager 사용 설명서의 공유 연락처 및 대응 계획 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteResourcePolicy의 섹션을 참조하세요. AWS CLI

### delete-response-plan

다음 코드 예시에서는 delete-response-plan을 사용하는 방법을 보여 줍니다.

AWS CLI

대응 계획을 삭제하려면

다음 delete-response-plan 예제에서는 지정된 대응 계획을 삭제합니다.

```
aws ssm-incidents delete-response-plan \
    --arn "arn:aws:ssm-incidents::111122223333:response-plan/example-response"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Incident Manager 사용 설명서의 인시던트 준비를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteResponsePlan의 섹션을 참조하세요. AWS CLI

### delete-timeline-event

다음 코드 예시에서는 delete-timeline-event을 사용하는 방법을 보여 줍니다.

AWS CLI

타임라인 이벤트를 삭제하려면

다음 delete-timeline-event 예제에서는 지정된 인시던트 레코드에서 사용자 지정 타임라인 이벤트를 삭제합니다.

```
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"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Incident Manager 사용 설명서의 인시던트 세부 정보를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteTimelineEvent의 섹션을 참조하세요. AWS CLI

### get-incident-record

다음 코드 예시에서는 get-incident-record을 사용하는 방법을 보여 줍니다.

AWS CLI

인시던트 레코드를 가져오려면

다음 get-incident-record 예제에서는 지정된 인시던트 레코드에 대한 세부 정보를 가져옵니다.

```
aws ssm-incidents get-incident-record \
    --arn "arn:aws:ssm-incidents::111122223333:incident-record/Example-Response-
Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Incident Manager 사용 설명서의 인시던트 세부 정보를 참조하세요.

- 자세한 API 내용은 명령 참조GetIncidentRecord의 섹션을 참조하세요. AWS CLI

### get-replication-set

다음 코드 예시에서는 get-replication-set을 사용하는 방법을 보여 줍니다.

AWS CLI

복제 세트를 가져오려면

다음 get-replication-set 예제에서는 Incident Manager가 Amazon Web Services 계정의 데이터를 복제하고 암호화하는 데 사용하는 복제 세트의 세부 정보를 가져옵니다.

```
aws ssm-incidents get-replication-set \
    --arn "arn:aws:ssm-incidents::111122223333:replication-set/c4bcb603-4bf9-bb3f-413c-08df53673b57"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Incident Manager 사용 설명서의 Incident Manager 복제 세트 사용을 참조하세요.

- 자세한 API 내용은 명령 참조GetReplicationSet의 섹션을 참조하세요. AWS CLI

### get-resource-policies

다음 코드 예시에서는 get-resource-policies을 사용하는 방법을 보여 줍니다.

AWS CLI

대응 계획에 대한 리소스 정책을 나열하려면

다음 command-name 예제에서는 지정된 대응 계획과 연결된 리소스 정책을 나열합니다.

```
aws ssm-incidents get-resource-policies \
--resource-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan"
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Incident Manager 사용 설명서의 공유 연락처 및 대응 계획 작업을 참조하세요.

- 자세한 API 내용은 명령 참조GetResourcePolicies의 섹션을 참조하세요. AWS CLI

### get-response-plan

다음 코드 예시에서는 get-response-plan을 사용하는 방법을 보여 줍니다.

AWS CLI

대응 계획의 세부 정보를 가져오려면

다음 command-name 예제에서는 AWS 계정의 지정된 대응 계획에 대한 세부 정보를 가져옵니다.

```
aws ssm-incidents get-response-plan \
    --arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan"
```

출력:

```
{
    "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"
}
```

자세한 내용은 Incident Manager 사용 설명서의 인시던트 준비를 참조하세요.

- 자세한 API 내용은 명령 참조GetResponsePlan의 섹션을 참조하세요. AWS CLI

### get-timeline-event

다음 코드 예시에서는 get-timeline-event을 사용하는 방법을 보여 줍니다.

AWS CLI

타임라인 이벤트의 세부 정보를 가져오려면

다음 get-timeline-event 예제에서는 지정된 타임라인 이벤트의 세부 정보를 반환합니다.

```
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"
```

출력:

```
{
    "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"
```

```
    }
}
```

자세한 내용은 Incident Manager 사용 설명서의 인시던트 세부 정보를 참조하세요.

- 자세한 API 내용은 명령 참조GetTimelineEvent의 섹션을 참조하세요. AWS CLI

### `list-incident-records`

다음 코드 예시에서는 `list-incident-records`을 사용하는 방법을 보여 줍니다.

AWS CLI

인시던트 레코드를 나열하려면

다음 `command-name` 예제에서는 Amazon Web Services 계정의 인시던트 레코드를 나열합니다.

```
aws ssm-incidents list-incident-records
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Incident Manager 사용 설명서의 인시던트 목록을 참조하세요.

- 자세한 API 내용은 명령 참조ListIncidentRecords의 섹션을 참조하세요. AWS CLI

### list-related-items

다음 코드 예시에서는 list-related-items을 사용하는 방법을 보여 줍니다.

AWS CLI

관련 항목을 나열하려면

다음 list-related-items 예제에서는 지정된 인시던트의 관련 항목을 나열합니다.

```
aws ssm-incidents list-related-items \
    --incident-record-arn "arn:aws:ssm-incidents::111122223333:incident-record/
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Incident Manager 사용 설명서의 인시던트 세부 정보를 참조하세요.

- 자세한 API 내용은 명령 참조ListRelatedItems의 섹션을 참조하세요. AWS CLI

### `list-replication-sets`

다음 코드 예시에서는 `list-replication-sets`을 사용하는 방법을 보여 줍니다.

AWS CLI

복제 세트를 나열하려면

다음 `list-replication-set` 예제에서는 Incident Manager가 AWS 계정의 데이터를 복제하고 암호화하는 데 사용하는 복제 세트를 나열합니다.

```
aws ssm-incidents list-replication-sets
```

출력:

```
{
    "replicationSetArns": [
        "arn:aws:ssm-incidents::111122223333:replication-set/c4bcb603-4bf9-
bb3f-413c-08df53673b57"
    ]
}
```

자세한 내용은 Incident Manager 사용 설명서의 Incident Manager 복제 세트 사용을 참조하세요.

- 자세한 API 내용은 명령 참조ListReplicationSets의 섹션을 참조하세요. AWS CLI

### `list-response-plans`

다음 코드 예시에서는 `list-response-plans`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 대응 계획을 나열하려면

다음 `list-response-plans` 예제에서는 Amazon Web Services 계정에서 사용 가능한 응답 계획을 나열합니다.

```
aws ssm-incidents list-response-plans
```

출력:

```
{
    "responsePlanSummaries": [
        {
            "arn": "arn:aws:ssm-incidents::111122223333:response-plan/Example-
Response-Plan",
            "displayName": "Example response plan",
            "name": "Example-Response-Plan"
        }
    ]
}
```

자세한 내용은 Incident Manager 사용 설명서의 인시던트 준비를 참조하세요.

- 자세한 API 내용은 명령 참조ListResponsePlans의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

대응 계획의 태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 지정된 응답 계획과 연결된 태그를 나열합니다.

```
aws ssm-incidents list-tags-for-resource \
    --resource-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-
Response-Plan"
```

출력:

```
{
    "tags": {
    "group1": "1"
    }
}
```

자세한 내용은 Incident Manager 사용 설명서의 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### list-timeline-events

다음 코드 예시에서는 list-timeline-events을 사용하는 방법을 보여 줍니다.

AWS CLI

인시던트의 타임라인 이벤트를 나열하려면

다음 command-name 예제에서는 지정된 인시던트의 타임라인 이벤트를 나열합니다.

```
aws ssm-incidents list-timeline-events \
    --incident-record-arn "arn:aws:ssm-incidents::111122223333:incident-record/
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Incident Manager 사용 설명서의 인시던트 세부 정보를 참조하세요.

- 자세한 API 내용은 명령 참조ListTimelineEvents의 섹션을 참조하세요. AWS CLI

### put-resource-policy

다음 코드 예시에서는 put-resource-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

대응 계획 및 인시던트를 공유하려면

다음 command-name 예제에서는 지정된 보안 주체와 대응 계획 및 관련 인시던트를 공유하는 리소스 정책을 Example-Response-Plan 에 추가합니다.

```
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::222333334444: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/*\"]}]}"
```

출력:

```
{
    "policyId": "be8b57191f0371f1c6827341aa3f0a03"
}
```

자세한 내용은 Incident Manager 사용 설명서의 공유 연락처 및 대응 계획 작업을 참조하세요.

- 자세한 API 내용은 명령 참조PutResourcePolicy의 섹션을 참조하세요. AWS CLI

### `start-incident`

다음 코드 예시에서는 `start-incident`을 사용하는 방법을 보여 줍니다.

AWS CLI

인시던트를 시작하려면

다음 `start-incident` 예제에서는 지정된 대응 계획을 사용하여 인시던트를 시작합니다.

```
aws ssm-incidents start-incident \
    --response-plan-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-
Response-Plan"
```

출력:

```
{
    "incidentRecordArn": "arn:aws:ssm-incidents::682428703967:incident-record/
Example-Response-Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308"
}
```

자세한 내용은 Incident Manager 사용 설명서의 인시던트 생성을 참조하세요.

- 자세한 API 내용은 명령 참조StartIncident의 섹션을 참조하세요. AWS CLI

### `tag-resource`

다음 코드 예시에서는 `tag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

대응 계획에 태그를 지정하려면

다음 `tag-resource` 예제에서는 제공된 태그 키-값 페어로 지정된 응답 계획에 태그를 지정합니다.

```
aws ssm-incidents tag-resource \
    --resource-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan" \
    --tags '{"group1":"1"}'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Incident Manager 사용 설명서의 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### `untag-resource`

다음 코드 예시에서는 `untag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

대응 계획에서 태그를 제거하려면

다음 `untag-resource` 예제에서는 응답 계획에서 지정된 태그를 제거합니다.

```
aws ssm-incidents untag-resource \
    --resource-arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan" \
    --tag-keys '["group1"]'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Incident Manager 사용 설명서의 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-deletion-protection

다음 코드 예시에서는 update-deletion-protection을 사용하는 방법을 보여 줍니다.

AWS CLI

복제 세트 삭제 방지를 업데이트하려면

다음 update-deletion-protection 예제에서는 복제 세트의 마지막 리전을 삭제하지 않도록 계정의 삭제 보호를 업데이트합니다.

```
aws ssm-incidents update-deletion-protection \
    --arn "arn:aws:ssm-incidents::111122223333:replication-set/
a2bcc5c9-0f53-8047-7fef-c20749989b40" \
    --deletion-protected
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Incident Manager 사용 설명서의 Incident Manager 복제 세트 사용을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateDeletionProtection의 섹션을 참조하세요. AWS CLI

### update-incident-record

다음 코드 예시에서는 update-incident-record을 사용하는 방법을 보여 줍니다.

AWS CLI

인시던트 레코드를 업데이트하려면

다음 command-name 예제에서는 지정된 인시던트를 해결합니다.

```
aws ssm-incidents update-incident-record \
    --arn "arn:aws:ssm-incidents::111122223333:incident-record/Example-Response-
Plan/6ebcc812-85f5-b7eb-8b2f-283e4d844308" \
    --status "RESOLVED"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Incident Manager 사용 설명서의 인시던트 세부 정보를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateIncidentRecord의 섹션을 참조하세요. AWS CLI

### update-related-items

다음 코드 예시에서는 update-related-items을 사용하는 방법을 보여 줍니다.

AWS CLI

인시던트 관련 항목을 업데이트하려면

다음 update-related-item 예제에서는 지정된 인시던트 레코드에서 관련 항목을 제거합니다.

```
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"}}}'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Incident Manager 사용 설명서의 인시던트 세부 정보를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateRelatedItems의 섹션을 참조하세요. AWS CLI

### update-replication-set

다음 코드 예시에서는 update-replication-set을 사용하는 방법을 보여 줍니다.

AWS CLI

복제 세트를 업데이트하려면

다음 command-name 예제에서는 복제 세트에서 us-east-2 리전을 삭제합니다.

```
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"}}]'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Incident Manager 사용 설명서의 Incident Manager 복제 세트 사용을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateReplicationSet의 섹션을 참조하세요. AWS CLI

### update-response-plan

다음 코드 예시에서는 update-response-plan을 사용하는 방법을 보여 줍니다.

AWS CLI

대응 계획을 업데이트하려면

다음 update-response-plan 예제에서는 지정된 응답 계획에서 채팅 채널을 제거합니다.

```
aws ssm-incidents update-response-plan \
    --arn "arn:aws:ssm-incidents::111122223333:response-plan/Example-Response-Plan"
 \
    --chat-channel '{"empty":{}}'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Incident Manager 사용 설명서의 인시던트 준비를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateResponsePlan의 섹션을 참조하세요. AWS CLI

### update-timeline-event

다음 코드 예시에서는 update-timeline-event을 사용하는 방법을 보여 줍니다.

AWS CLI

타임라인 이벤트를 업데이트하려면

다음 update-timeline-event 예제에서는 이벤트가 발생한 시간을 업데이트합니다.

```
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"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Incident Manager 사용 설명서의 인시던트 세부 정보를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateTimelineEvent의 섹션을 참조하세요. AWS CLI

## 를 사용한 Incident Manager Contacts 예제 AWS CLI

다음 코드 예제에서는 Incident Manager Contacts와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `accept-page`

다음 코드 예시에서는 `accept-page`을 사용하는 방법을 보여 줍니다.

AWS CLI

및 참여 중에 페이지를 수락하려면

다음 `accept-page` 예제에서는 고객 응대 채널로 전송된 수락 코드를 사용하여 페이지를 수락합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Incident Manager 사용 설명서의 연락처를 참조하세요.

- 자세한 API 내용은 명령 참조AcceptPage의 섹션을 참조하세요. AWS CLI

### `activate-contact-channel`

다음 코드 예시에서는 `activate-contact-channel`을 사용하는 방법을 보여 줍니다.

AWS CLI

연락처의 연락 채널 활성화

다음 `activate-contact-channel` 예제에서는 연락 채널을 활성화하고 인시던트의 일부로 사용할 수 있도록 합니다.

```
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"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Incident Manager 사용 설명서의 연락처를 참조하세요.

- 자세한 API 내용은 명령 참조ActivateContactChannel의 섹션을 참조하세요. AWS CLI

### `command-name`

다음 코드 예시에서는 `command-name`을 사용하는 방법을 보여 줍니다.

AWS CLI

연락처를 삭제하려면

다음 `command-name` 예제에서는 연락처를 삭제합니다. 연락은 더 이상 해당 연락과 관련된 에스컬레이션 계획에서 연결할 수 없습니다.

```
aws ssm-contacts delete-contact \
    --contact-id "arn:aws:ssm-contacts:us-east-1:682428703967:contact/alejr"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Incident Manager 사용 설명서의 연락처를 참조하세요.

- 자세한 API 내용은 명령 참조CommandName의 섹션을 참조하세요. AWS CLI

### create-contact-channel

다음 코드 예시에서는 create-contact-channel을 사용하는 방법을 보여 줍니다.

AWS CLI

고객 응대 채널을 생성하려면

연락 Akua Mansa에 SMS 대한 유형의 연락 채널을 생성합니다. 연락 채널은 유형 SMS, EMAIL또는 로 생성할 수 있습니다VOICE.

```
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"}'
```

출력:

```
{
    "ContactChannelArn": "arn:aws:ssm-contacts:us-east-1:111122223333:contact-
channel/akuam/02f506b9-ea5d-4764-af89-2daa793ff024"
}
```

자세한 내용은 Incident Manager 사용 설명서의 연락처를 참조하세요.

- 자세한 API 내용은 명령 참조CreateContactChannel의 섹션을 참조하세요. AWS CLI

### create-contact

다음 코드 예시에서는 create-contact을 사용하는 방법을 보여 줍니다.

AWS CLI

연락처를 생성하려면

다음 create-contact 예제에서는 빈 계획으로 환경에 연락처를 생성합니다. 연락 채널을 생성한 후 계획을 업데이트할 수 있습니다. ARN 이 명령의 출력과 함께 명령을 사용합니다 create-contact-channel. 이 연락에 대한 연락 채널을 생성한 후 업데이트 연락을 사용하여 계획을 업데이트합니다.

```
aws ssm-contacts create-contact \
    --alias "akuam" \
    --display-name "Akua Mansa" \
```

```
    --type PERSONAL \
    --plan '{"Stages": []}'
```

출력:

```
{
    "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam"
}
```

자세한 내용은 Incident Manager 사용 설명서의 연락처를 참조하세요.

- 자세한 API 내용은 명령 참조CreateContact의 섹션을 참조하세요. AWS CLI

### deactivate-contact-channel

다음 코드 예시에서는 deactivate-contact-channel을 사용하는 방법을 보여 줍니다.

AWS CLI

연락 채널을 비활성화하려면

다음 deactivate-contact-channel 예제에서는 연락 채널을 비활성화합니다. 고객 응대 채널을 비활성화하면 인시던트 발생 시 고객 응대 채널이 더 이상 호출되지 않습니다. 언제든지 activate-contact-channel 명령을 사용하여 연락 채널을 다시 활성화할 수도 있습니다.

```
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"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Incident Manager 사용 설명서의 연락처를 참조하세요.

- 자세한 API 내용은 명령 참조DeactivateContactChannel의 섹션을 참조하세요. AWS CLI

### delete-contact-channel

다음 코드 예시에서는 delete-contact-channel을 사용하는 방법을 보여 줍니다.

AWS CLI

연락 채널을 삭제하려면

다음 delete-contact-channel 예제에서는 연락 채널을 삭제합니다. 연락 채널을 삭제하면 인시던트 발생 시 연락 채널이 호출되지 않습니다.

```
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"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Incident Manager 사용 설명서의 연락처를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteContactChannel의 섹션을 참조하세요. AWS CLI

### delete-contact

다음 코드 예시에서는 delete-contact을 사용하는 방법을 보여 줍니다.

AWS CLI

연락처를 삭제하려면

다음 delete-contact 예제에서는 연락처를 삭제합니다. 연락은 더 이상 해당 연락과 관련된 에스컬레이션 계획에서 연결할 수 없습니다.

```
aws ssm-contacts delete-contact \
    --contact-id "arn:aws:ssm-contacts:us-east-1:111122223333:contact/alejr"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Incident Manager 사용 설명서의 연락처를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteContact의 섹션을 참조하세요. AWS CLI

### describe-engagement

다음 코드 예시에서는 describe-engagement을 사용하는 방법을 보여 줍니다.

AWS CLI

참여의 세부 정보를 설명하려면

다음 describe-engagement 예제에서는 고객 응대 또는 에스컬레이션 계획에 대한 참여 세부 정보를 나열합니다. 주제와 콘텐츠가 연락 채널로 전송됩니다.

```
aws ssm-contacts describe-engagement \
    --engagement-id "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/
example_escalation/69e40ce1-8dbb-4d57-8962-5fbe7fc53356"
```

출력:

```
{
    "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"
}
```

자세한 내용은 Incident Manager 사용 설명서의 연락처를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeEngagement의 섹션을 참조하세요. AWS CLI

### describe-page

다음 코드 예시에서는 describe-page을 사용하는 방법을 보여 줍니다.

AWS CLI

연락처 채널에 대한 페이지 세부 정보를 나열하려면

다음 describe-page 예제에서는 연락 채널에 대한 페이지의 세부 정보를 나열합니다. 페이지에는 제공된 주제와 콘텐츠가 포함됩니다.

```
aws ssm-contacts describe-page \
    --page-id "arn:aws:ssm-contacts:us-east-2:111122223333:page/akuam/ad0052bd-
e606-498a-861b-25726292eb93"
```

출력:

```
{
```

```
    "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"
}
```

자세한 내용은 Incident Manager 사용 설명서의 연락처를 참조하세요.

- 자세한 API 내용은 명령 참조DescribePage의 섹션을 참조하세요. AWS CLI

### `get-contact-channel`

다음 코드 예시에서는 `get-contact-channel`을 사용하는 방법을 보여 줍니다.

AWS CLI

연락 채널의 세부 정보를 나열하려면

다음 `get-contact-channel` 예제에서는 연락 채널의 세부 정보를 나열합니다.

```
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"
```

출력:

```
{
    "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"
}
```

자세한 내용은 Incident Manager 사용 설명서의 연락처를 참조하세요.

- 자세한 API 내용은 명령 참조GetContactChannel의 섹션을 참조하세요. AWS CLI

### get-contact-policy

다음 코드 예시에서는 get-contact-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

연락처의 리소스 정책을 나열하려면

다음 get-contact-policy 예제에서는 지정된 연락처와 연결된 리소스 정책을 나열합니다.

```
aws ssm-contacts get-contact-policy \
    --contact-arn "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam"
```

출력:

```
{
    "ContactArn": "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam",
    "Policy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Sid\":
\"SharePolicyForDocumentationDralia\",\"Effect\":\"Allow\",\"Principal\":
{\"AWS\":\"222233334444\"},\"Action\":[\"ssm-contacts:GetContact\",\"ssm-
contacts:StartEngagement\",\"ssm-contacts:DescribeEngagement\",\"ssm-
contacts:ListPagesByEngagement\",\"ssm-contacts:StopEngagement\"],\"Resource
\":[\"arn:aws:ssm-contacts:*:111122223333:contact/akuam\",\"arn:aws:ssm-
contacts:*:111122223333:engagement/akuam/*\"]}]}"
}
```

자세한 내용은 Incident Manager 사용 설명서의 공유 연락처 및 대응 계획 작업을 참조하세요.

- 자세한 API 내용은 명령 참조GetContactPolicy의 섹션을 참조하세요. AWS CLI

### get-contact

다음 코드 예시에서는 get-contact을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 고객 응대 계획 설명

다음 get-contact 예제에서는 연락처에 대해 설명합니다.

```
aws ssm-contacts get-contact \
    --contact-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam"
```

출력:

```
{
    "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
                        }
                    }
                ]
            }
        ]
    }
}
```

예제 2: 에스컬레이션 계획 설명

다음 get-contact 예제에서는 에스컬레이션 계획을 설명합니다.

```
aws ssm-contacts get-contact \
--contact-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
example_escalation"
```

출력:

```
{
    "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
                        }
                    }
                ]
            }
        ]
    }
}
```

자세한 내용은 Incident Manager 사용 설명서의 연락처를 참조하세요.

- 자세한 API 내용은 명령 참조GetContact의 섹션을 참조하세요. AWS CLI

### `list-contact-channels`

다음 코드 예시에서는 `list-contact-channels`을 사용하는 방법을 보여 줍니다.

AWS CLI

연락처의 연락처 채널을 나열하려면

다음 `list-contact-channels` 예제에서는 지정된 연락의 사용 가능한 연락 채널을 나열합니다.

```
aws ssm-contacts list-contact-channels \
    --contact-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam"
```

출력:

```
{
    [
        {
            "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"
        }
    ]
}
```

자세한 내용은 Incident Manager 사용 설명서의 연락처를 참조하세요.

- 자세한 API 내용은 명령 참조ListContactChannels의 섹션을 참조하세요. AWS CLI

### `list-contacts`

다음 코드 예시에서는 `list-contacts`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 에스컬레이션 계획 및 연락처를 나열하려면

다음 `list-contacts` 예제에서는 계정의 연락 및 에스컬레이션 계획을 나열합니다.

```
aws ssm-contacts list-contacts
```

출력:

```
{
    "Contacts": [
        {
            "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
akuam",
            "Alias": "akuam",
            "DisplayName": "Akua Mansa",
            "Type": "PERSONAL"
        },
        {
            "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
alejr",
            "Alias": "alejr",
            "DisplayName": "Alejandro Rosalez",
            "Type": "PERSONAL"
        },
        {
            "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
anasi",
            "Alias": "anasi",
            "DisplayName": "Ana Carolina Silva",
            "Type": "PERSONAL"
        },
        {
            "ContactArn": "arn:aws:ssm-contacts:us-east-2:111122223333:contact/
example_escalation",
            "Alias": "example_escalation",
            "DisplayName": "Example Escalation",
            "Type": "ESCALATION"
        }
    ]
}
```

자세한 내용은 Incident Manager 사용 설명서의 연락처를 참조하세요.

- 자세한 API 내용은 명령 참조ListContacts의 섹션을 참조하세요. AWS CLI

### list-engagements

다음 코드 예시에서는 list-engagements을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 참여를 나열하려면

다음 list-engagements 예제에서는 에스컬레이션 계획 및 연락처에 대한 참여를 나열합니다. 단일 인시던트에 대한 참여를 나열할 수도 있습니다.

```
aws ssm-contacts list-engagements
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Incident Manager 사용 설명서의 연락처를 참조하세요.

- 자세한 API 내용은 명령 참조ListEngagements의 섹션을 참조하세요. AWS CLI

### `list-page-receipts`

다음 코드 예시에서는 `list-page-receipts`을 사용하는 방법을 보여 줍니다.

AWS CLI

페이지 수신을 나열하려면

다음 command-name 예제에서는 연락처가 페이지를 수신했는지 여부를 나열합니다.

```
aws ssm-contacts list-page-receipts \
    --page-id "arn:aws:ssm-contacts:us-east-2:111122223333:page/
akuam/94ea0c7b-56d9-46c3-b84a-a37c8b067ad3"
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Incident Manager 사용 설명서의 연락처를 참조하세요.

- 자세한 API 내용은 명령 참조ListPageReceipts의 섹션을 참조하세요. AWS CLI

### `list-pages-by-contact`

다음 코드 예시에서는 `list-pages-by-contact`을 사용하는 방법을 보여 줍니다.

AWS CLI

연락처별로 페이지를 나열하려면

다음 `list-pages-by-contact` 예제에서는 지정된 연락처의 모든 페이지를 나열합니다.

```
aws ssm-contacts list-pages-by-contact \
    --contact-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam"
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Incident Manager 사용 설명서의 연락처를 참조하세요.

- 자세한 API 내용은 명령 참조ListPagesByContact의 섹션을 참조하세요. AWS CLI

### `list-pages-by-engagement`

다음 코드 예시에서는 `list-pages-by-engagement`을 사용하는 방법을 보여 줍니다.

AWS CLI

참여에서 시작된 연락 채널 페이지를 나열합니다.

다음 `list-pages-by-engagement` 예제에서는 정의된 참여 계획을 사용하는 동안 발생한 페이지를 나열합니다.

```
aws ssm-contacts list-pages-by-engagement \
    --engagement-id "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/
akuam/78a29753-3674-4ac5-9f83-0468563567f0"
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Incident Manager 사용 설명서의 연락처를 참조하세요.

- 자세한 API 내용은 명령 참조ListPagesByEngagement의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

연락처의 태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 지정된 연락처의 태그를 나열합니다.

```
aws ssm-contacts list-tags-for-resource \
    --resource-arn "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam"
```

출력:

```
{
    "Tags": [
        {
            "Key": "group1",
            "Value": "1"
        }
    ]
}
```

자세한 내용은 Incident Manager 사용 설명서의 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### `put-contact-policy`

다음 코드 예시에서는 `put-contact-policy`을 사용하는 방법을 보여 줍니다.

AWS CLI

고객 응대 및 참여를 공유하려면

다음 `put-contact-policy` 예제에서는 보안 주체와 연락처 및 관련 참여를 공유하는 리소스 정책을 연락처 Akua에 추가합니다.

```
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\/*\"]}]}"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Incident Manager 사용 설명서의 공유 연락처 및 대응 계획 작업을 참조하세요.

- 자세한 API 내용은 명령 참조PutContactPolicy의 섹션을 참조하세요. AWS CLI

### `send-activation-code`

다음 코드 예시에서는 `send-activation-code`을 사용하는 방법을 보여 줍니다.

AWS CLI

활성화 코드를 보내려면

다음 `send-activation-code` 예제에서는 활성화 코드와 메시지를 지정된 연락 채널로 보냅니다.

```
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"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Incident Manager 사용 설명서의 연락처를 참조하세요.

- 자세한 API 내용은 명령 참조SendActivationCode의 섹션을 참조하세요. AWS CLI

### `start-engagement`

다음 코드 예시에서는 `start-engagement`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 연락처의 연락 채널을 호출하려면

다음 start-engagement 페이지는 연락처의 연락처 채널입니다. 발신자, 제목, 퍼블릭 주제 및 퍼블릭 콘텐츠에는 필드가 없습니다. Incident Manager는 제공된 채널 VOICE 또는 EMAIL 연락 채널로 주제와 콘텐츠를 전송합니다. Incident Manager는 제공된 SMS 연락 채널로 퍼블릭 주제 및 퍼블릭 콘텐츠를 전송합니다. 발신자는 누가 참여를 시작했는지 추적하는 데 사용됩니다.

```
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"
```

출력:

```
{
    "EngagementArn": "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/
akuam/607ced0e-e8fa-4ea7-8958-a237b8803f8f"
}
```

자세한 내용은 Incident Manager 사용 설명서의 연락처를 참조하세요.

예제 2: 제공된 에스컬레이션 계획에서 연락처를 호출합니다.

다음은 에스컬레이션 계획을 통해 고객 응대에 start-engagement 관여합니다. 각 연락처는 참여 계획에 따라 페이지가 지정됩니다.

```
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"
```

출력:

```
{
    "EngagementArn": "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/
example_escalation/69e40ce1-8dbb-4d57-8962-5fbe7fc53356"
```

```
}
```

자세한 내용은 Incident Manager 사용 설명서의 연락처를 참조하세요.

- 자세한 API 내용은 명령 참조StartEngagement의 섹션을 참조하세요. AWS CLI

### stop-engagement

다음 코드 예시에서는 stop-engagement을 사용하는 방법을 보여 줍니다.

AWS CLI

참여를 중지하려면

다음 stop-engagement 예제에서는 참여로 인해 추가 연락 및 연락 채널 페이징이 중지됩니다.

```
aws ssm-contacts stop-engagement \
    --engagement-id "arn:aws:ssm-contacts:us-east-2:111122223333:engagement/
example_escalation/69e40ce1-8dbb-4d57-8962-5fbe7fc53356"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Incident Manager 사용 설명서의 연락처를 참조하세요.

- 자세한 API 내용은 명령 참조StopEngagement의 섹션을 참조하세요. AWS CLI

### tag-resource

다음 코드 예시에서는 tag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

연락처에 태그를 지정하려면

다음 tag-resource 예제에서는 지정된 연락처에 제공된 태그 키 값 페어를 태그 지정합니다.

```
aws ssm-contacts tag-resource \
    --resource-arn "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam" \
    --tags '[{"Key":"group1","Value":"1"}]'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Incident Manager 사용 설명서의 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### untag-resource

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

연락처에서 태그를 제거하려면

다음 untag-resource 예제에서는 지정된 연락처에서 group1 태그를 제거합니다.

```
aws ssm-contacts untag-resource \
    --resource-arn "arn:aws:ssm-contacts:us-east-1:111122223333:contact/akuam" \
    --tag-keys "group1"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Incident Manager 사용 설명서의 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-contact-channel

다음 코드 예시에서는 update-contact-channel을 사용하는 방법을 보여 줍니다.

AWS CLI

연락 채널을 업데이트하려면

다음 update-contact-channel 예제에서는 연락 채널의 이름과 전송 주소를 업데이트합니다.

```
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"}'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Incident Manager 사용 설명서의 연락처를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateContactChannel의 섹션을 참조하세요. AWS CLI

### `update-contact`

다음 코드 예시에서는 `update-contact`을 사용하는 방법을 보여 줍니다.

AWS CLI

연락 참여 계획을 업데이트하려면

다음 `update-contact` 예제에서는 세 가지 유형의 연락 채널을 포함하도록 연락 Akua의 참여 계획을 업데이트합니다. 이는 Akua에 대한 연락 채널을 생성한 후 수행됩니다.

```
aws ssm-contacts update-contact \
    --contact-id "arn:aws:ssm-contacts:us-east-2:111122223333:contact/akuam" \
    --plan '{"Stages": [{"DurationInMinutes": 5, "Targets": [{"ChannelTargetInfo":
 {"ContactChannelId": "arn:aws:ssm-contacts:us-east-2:111122223333:contact-
channel/akuam/beb25840-5ac8-4644-95cc-7a8de390fa65","RetryIntervalInMinutes":
 1 }}]}, {"DurationInMinutes": 5, "Targets": [{"ChannelTargetInfo":
{"ContactChannelId": "arn:aws:ssm-contacts:us-east-2:111122223333:contact-channel/
akuam/49f3c24d-5f9f-4638-ae25-3f49e04229ad", "RetryIntervalInMinutes": 1}}]},
 {"DurationInMinutes": 5, "Targets": [{"ChannelTargetInfo": {"ContactChannelId":
 "arn:aws:ssm-contacts:us-east-2:111122223333:contact-channel/akuam/77d4f447-
f619-4954-afff-85551e369c2a","RetryIntervalInMinutes": 1 }}]}]}'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Incident Manager 사용 설명서의 연락처를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateContact의 섹션을 참조하세요. AWS CLI

## 를 사용한 Amazon Inspector 예제 AWS CLI

다음 코드 예제에서는 Amazon Inspector 와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### add-attributes-to-findings

다음 코드 예시에서는 add-attributes-to-findings을 사용하는 방법을 보여 줍니다.

AWS CLI

결과에 속성을 추가하려면

다음 add-attribute-to-finding 명령은 키Example가 이고 값이 인 속성을 ARN의 를 사용하여 example 결과에 할당합니다arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-8l1VIE0D/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/0-8l1VIE0D/run/0-Z02cjjug/finding/0-T8yM9mEU --attributes key=Example,value=example
```

출력:

```
{
    "failedItems": {}
}
```

자세한 내용은 Amazon Inspector 안내서의 Amazon Inspector 조사 결과를 참조하세요.

- 자세한 API 내용은 명령 참조AddAttributesToFindings의 섹션을 참조하세요. AWS CLI

### create-assessment-target

다음 코드 예시에서는 create-assessment-target을 사용하는 방법을 보여 줍니다.

AWS CLI

평가 대상을 생성하려면

다음 create-assessment-target 명령은 ARN 의 를 사용하여 리소스 그룹을 ExampleAssessmentTarget 사용하여 라는 평가 대상을 생성합니다arn: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
```

출력:

```
{
    "assessmentTargetArn": "arn:aws:inspector:us-west-2:123456789012:target/0-
nvgVhaxX"
}
```

자세한 내용은 Amazon Inspector 가이드의 Amazon Inspector 평가 대상을 참조하세요.

- 자세한 API 내용은 명령 참조CreateAssessmentTarget의 섹션을 참조하세요. AWS CLI

### create-assessment-template

다음 코드 예시에서는 create-assessment-template을 사용하는 방법을 보여 줍니다.

AWS CLI

평가 템플릿을 생성하려면

다음 create-assessment-template 명령은 ARN 의 를 사용하여 평가 대상에 ExampleAssessmentTemplate 대해 라는 평가 템플릿을 생성합니다arn: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
```

출력:

```
{
```

```
    "assessmentTemplateArn": "arn:aws:inspector:us-west-2:123456789012:target/0-
nvgVhaxX/template/0-it5r2S4T"
}
```

자세한 내용은 Amazon Inspector 가이드의 Amazon Inspector 평가 템플릿 및 평가 실행을 참조하세요.

- 자세한 API 내용은 명령 참조CreateAssessmentTemplate의 섹션을 참조하세요. AWS CLI

### `create-filter`

다음 코드 예시에서는 `create-filter`을 사용하는 방법을 보여 줍니다.

AWS CLI

필터를 생성하려면

다음 `create-filter` 예제에서는 ECR 인스턴스 유형 조사 결과를 생략하는 억제 규칙을 생성합니다.

```
aws inspector2 create-filter \
    --name "ExampleSuppressionRuleECR" \
    --description "This suppression rule omits ECR instance type findings" \
    --action SUPPRESS \
    --filter-criteria 'resourceType=[{comparison="EQUALS",
 value="AWS_ECR_INSTANCE"}]'
```

출력:

```
{
    "arn": "arn:aws:inspector2:us-west-2:123456789012:owner/o-EXAMPLE222/filter/
EXAMPLE444444444"
}
```

자세한 내용은 Amazon Inspector 사용 설명서의 Amazon Inspector 조사 결과 필터링을 참조하세요. Amazon Inspector

- 자세한 API 내용은 명령 참조CreateFilter의 섹션을 참조하세요. AWS CLI

### `create-findings-report`

다음 코드 예시에서는 `create-findings-report`을 사용하는 방법을 보여 줍니다.

AWS CLI

조사 결과 보고서를 생성하려면

다음 create-findings-report 예제에서는 결과 보고서를 생성합니다.

```
aws inspector2 create-findings-report \
    --report-format CSV \
    --s3-destination bucketName=inspector-sbom-123456789012,keyPrefix=sbom-key,kmsKeyArn=arn:aws:kms:us-west-2:123456789012:key/a1b2c3d4-5678-90ab-cdef-EXAMPLE33333 \
    --filter-criteria '{"ecrImageRepositoryName":[{"comparison":"EQUALS","value":"debian"}]}'
```

출력:

```
{
    "reportId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333"
}
```

자세한 내용은 Amazon Inspector 사용 설명서의 Amazon Inspector에서 결과 관리를 참조하세요. Amazon Inspector

- 자세한 API 내용은 명령 참조CreateFindingsReport의 섹션을 참조하세요. AWS CLI

## create-resource-group

다음 코드 예시에서는 create-resource-group을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 그룹을 생성하려면

다음 create-resource-group 명령은 의 태그 키Name와 의 값을 사용하여 리소스 그룹을 생성합니다example.

```
aws inspector create-resource-group --resource-group-tags key=Name,value=example
```

출력:

```
{
```

```
    "resourceGroupArn": "arn:aws:inspector:us-west-2:123456789012:resourcegroup/0-
AB6DMKnv"
}
```

자세한 내용은 Amazon Inspector 가이드의 Amazon Inspector 평가 대상을 참조하세요.

- 자세한 API 내용은 명령 참조CreateResourceGroup의 섹션을 참조하세요. AWS CLI

### `create-sbom-export`

다음 코드 예시에서는 `create-sbom-export`을 사용하는 방법을 보여 줍니다.

AWS CLI

소프트웨어 재료 명세서(SBOM) 보고서를 생성하려면

다음 `create-sbom-export` 예제에서는 소프트웨어 재료표(SBOM) 보고서를 생성합니다.

```
aws inspector2 create-sbom-export \
    --report-format SPDX_2_3 \
    --resource-filter-criteria
 'ecrRepositoryName=[{comparison="EQUALS",value="debian"}]' \
    --s3-destination bucketName=inspector-sbom-123456789012,keyPrefix=sbom-
key,kmsKeyArn=arn:aws:kms:us-west-2:123456789012:key/a1b2c3d4-5678-90ab-cdef-
EXAMPLE33333
```

출력:

```
{
    "reportId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333"
}
```

자세한 내용은 Amazon Inspector 사용 설명서의 Amazon InspectorSBOMs로 내보내기를 참조하세요. Amazon Inspector

- 자세한 API 내용은 명령 참조CreateSbomExport의 섹션을 참조하세요. AWS CLI

### `delete-assessment-run`

다음 코드 예시에서는 `delete-assessment-run`을 사용하는 방법을 보여 줍니다.

AWS CLI

평가 실행을 삭제하려면

다음 delete-assessment-run 명령은 ARN 의 를 사용하여 평가 실행을 삭제합니다arn: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
```

자세한 내용은 Amazon Inspector 가이드의 Amazon Inspector 평가 템플릿 및 평가 실행을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteAssessmentRun의 섹션을 참조하세요. AWS CLI

### delete-assessment-target

다음 코드 예시에서는 delete-assessment-target을 사용하는 방법을 보여 줍니다.

AWS CLI

평가 대상을 삭제하려면

다음 delete-assessment-target 명령은 ARN 의 를 사용하여 평가 대상을 삭제합니다arn: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
```

자세한 내용은 Amazon Inspector 가이드의 Amazon Inspector 평가 대상을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteAssessmentTarget의 섹션을 참조하세요. AWS CLI

### delete-assessment-template

다음 코드 예시에서는 delete-assessment-template을 사용하는 방법을 보여 줍니다.

AWS CLI

평가 템플릿을 삭제하려면

다음 delete-assessment-template 명령은 ARN 의 를 사용하여 평가 템플릿을 삭제합니다arn: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
```

자세한 내용은 Amazon Inspector 가이드의 Amazon Inspector 평가 템플릿 및 평가 실행을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteAssessmentTemplate의 섹션을 참조하세요. AWS CLI

### delete-filter

다음 코드 예시에서는 delete-filter을 사용하는 방법을 보여 줍니다.

AWS CLI

필터를 삭제하려면

다음 delete-filter 예제에서는 필터를 삭제합니다.

```
aws inspector2 delete-filter \
    --arn "arn:aws:inspector2:us-west-2:123456789012:owner/o-EXAMPLE222/filter/
EXAMPLE444444444"
```

출력:

```
{
    "arn": "arn:aws:inspector2:us-west-2:123456789012:owner/o-EXAMPLE222/filter/
EXAMPLE444444444"
}
```

자세한 내용은 Amazon Inspector 사용 설명서의 Amazon Inspector 결과 필터링을 참조하세요. Amazon Inspector

- 자세한 API 내용은 명령 참조DeleteFilter의 섹션을 참조하세요. AWS CLI

### describe-assessment-runs

다음 코드 예시에서는 describe-assessment-runs을 사용하는 방법을 보여 줍니다.

AWS CLI

평가 실행을 설명하려면

다음 describe-assessment-run 명령은 ARN 의 를 사용한 평가 실행을 설명합니다arn: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
```

출력:

```
{
        "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": {}
}
```

자세한 내용은 Amazon Inspector 가이드의 Amazon Inspector 평가 템플릿 및 평가 실행을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeAssessmentRuns의 섹션을 참조하세요. AWS CLI

### `describe-assessment-targets`

다음 코드 예시에서는 `describe-assessment-targets`을 사용하는 방법을 보여 줍니다.

AWS CLI

평가 대상을 설명하려면

다음 describe-assessment-targets 명령은 ARN 의 를 사용하여 평가 대상을 설명합니다arn: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
```

출력:

```
{
      "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": {}
}
```

자세한 내용은 Amazon Inspector 가이드의 Amazon Inspector 평가 대상을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeAssessmentTargets의 섹션을 참조하세요. AWS CLI

### describe-assessment-templates

다음 코드 예시에서는 describe-assessment-templates을 사용하는 방법을 보여 줍니다.

AWS CLI

평가 템플릿을 설명하려면

다음 describe-assessment-templates 명령은 ARN 의 를 사용하여 평가 템플릿을 설명합니다arn: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
```

출력:

```
{
      "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": {}
}
```

자세한 내용은 Amazon Inspector 가이드의 Amazon Inspector 평가 템플릿 및 평가 실행을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeAssessmentTemplates의 섹션을 참조하세요. AWS CLI

### describe-cross-account-access-role

다음 코드 예시에서는 `describe-cross-account-access-role`을 사용하는 방법을 보여 줍니다.

AWS CLI

교차 계정 액세스 역할을 설명하려면

다음 `describe-cross-account-access-role` 명령은 Amazon Inspector가 AWS 계정에 액세스할 수 있도록 하는 IAM 역할을 설명합니다.

```
aws inspector describe-cross-account-access-role
```

출력:

```
{
        "registeredAt": 1458069182.826,
        "roleArn": "arn:aws:iam::123456789012:role/inspector",
```

```
        "valid": true
}
```

자세한 내용은 Amazon Inspector 안내서의 Amazon Inspector 설정을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeCrossAccountAccessRole의 섹션을 참조하세요. AWS CLI

### `describe-findings`

다음 코드 예시에서는 `describe-findings`을 사용하는 방법을 보여 줍니다.

AWS CLI

조사 결과를 설명하려면

다음 `describe-findings` 명령은 ARN 의 를 사용하여 결과를 설명합니다`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
```

출력:

```
{
      "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": []
        }
    ]
}
```

자세한 내용은 Amazon Inspector 가이드의 Amazon Inspector 조사 결과를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeFindings의 섹션을 참조하세요. AWS CLI

### describe-resource-groups

다음 코드 예시에서는 describe-resource-groups을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 그룹을 설명하려면

다음 describe-resource-groups 명령은 ARN 의 를 사용하여 리소스 그룹을 설명합니다arn: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
```

출력:

```
{
    "failedItems": {},
    "resourceGroups": [
      {
```

```
            "arn": "arn:aws:inspector:us-west-2:123456789012:resourcegroup/0-
PyGXopAI",
            "createdAt": 1458074191.098,
            "tags": [
              {
                    "key": "Name",
                    "value": "example"
              }
            ]
        }
      ]
}
```

자세한 내용은 Amazon Inspector 가이드의 Amazon Inspector 평가 대상을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeResourceGroups의 섹션을 참조하세요. AWS CLI

### `describe-rules-packages`

다음 코드 예시에서는 `describe-rules-packages`을 사용하는 방법을 보여 줍니다.

AWS CLI

규칙 패키지를 설명하려면

다음 `describe-rules-packages` 명령은 ARN 의 를 사용하여 규칙 패키지를 설명합니다`arn:aws:inspector:us-west-2:758058086616:rulespackage/0-9hgA516p`.

```
aws inspector describe-rules-packages --rules-package-arns arn:aws:inspector:us-
west-2:758058086616:rulespackage/0-9hgA516p
```

출력:

```
{
      "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"
            }
        ]
}
```

자세한 내용은 Amazon Inspector 가이드의 Amazon Inspector 규칙 패키지 및 규칙을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeRulesPackages의 섹션을 참조하세요. AWS CLI

### `get-configuration`

다음 코드 예시에서는 `get-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

Inspector 스캔에 대한 설정 구성을 가져오려면

다음 `get-configuration` 예제에서는 Inspector 스캔에 대한 설정 구성을 가져옵니다.

```
aws inspector2 get-configuration
```

출력:

```
{
    "ec2Configuration": {
        "scanModeState": {
            "scanMode": "EC2_HYBRID",
            "scanModeStatus": "SUCCESS"
        }
    },
    "ecrConfiguration": {
        "rescanDurationState": {
            "pullDateRescanDuration": "DAYS_90",
```

```
            "rescanDuration": "DAYS_30",
            "status": "SUCCESS",
            "updatedAt": "2024-05-14T21:16:20.237000+00:00"
        }
    }
}
```

자세한 내용은 Amazon Inspector 사용 설명서의 Amazon Inspector를 사용한 자동 리소스 스캔을 참조하세요. Amazon Inspector

- 자세한 API 내용은 명령 참조GetConfiguration의 섹션을 참조하세요. AWS CLI

### get-telemetry-metadata

다음 코드 예시에서는 get-telemetry-metadata을 사용하는 방법을 보여 줍니다.

AWS CLI

원격 측정 메타데이터를 가져오려면

다음 get-telemetry-metadata 명령은 ARN 의 를 사용하여 평가 실행을 위해 수집된 데이터에 대한 정보를 생성합니다arn: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
```

출력:

```
{
      "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": "InspectorLoadImageInProcess"
```

```
        }
      ]
}
```

- 자세한 API 내용은 명령 참조GetTelemetryMetadata의 섹션을 참조하세요. AWS CLI

### `list-account-permissions`

다음 코드 예시에서는 `list-account-permissions`을 사용하는 방법을 보여 줍니다.

AWS CLI

계정 권한을 나열하려면

다음 `list-account-permissions` 예제에서는 계정 권한을 나열합니다.

```
aws inspector2 list-account-permissions
```

출력:

```
{
    "permissions": [
        {
            "operation": "ENABLE_SCANNING",
            "service": "ECR"
        },
        {
            "operation": "DISABLE_SCANNING",
            "service": "ECR"
        },
        {
            "operation": "ENABLE_REPOSITORY",
            "service": "ECR"
        },
        {
            "operation": "DISABLE_REPOSITORY",
            "service": "ECR"
        },
        {
            "operation": "ENABLE_SCANNING",
            "service": "EC2"
        },
```

```
        {
            "operation": "DISABLE_SCANNING",
            "service": "EC2"
        },
        {
            "operation": "ENABLE_SCANNING",
            "service": "LAMBDA"
        },
        {
            "operation": "DISABLE_SCANNING",
            "service": "LAMBDA"
        }
    ]
}
```

자세한 내용은 Amazon Inspector 사용 설명서의 Amazon Inspector용 자격 증명 및 액세스 관리를 참조하세요. Amazon Inspector

- 자세한 API 내용은 명령 참조ListAccountPermissions의 섹션을 참조하세요. AWS CLI

### `list-assessment-run-agents`

다음 코드 예시에서는 `list-assessment-run-agents`을 사용하는 방법을 보여 줍니다.

AWS CLI

평가 실행 에이전트를 나열하려면

다음 `list-assessment-run-agents` 명령은 지정된 로 실행되는 평가의 에이전트를 나열합니다ARN.

```
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
```

출력:

```
{
    "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": "InspectorLoadImageInProcess"
                }
            ]
        }
    ]
}
```

자세한 내용은 Amazon Inspector 사용 설명서의 AWS 에이전트를 참조하세요.

- 자세한 API 내용은 명령 참조ListAssessmentRunAgents의 섹션을 참조하세요. AWS CLI

### `list-assessment-runs`

다음 코드 예시에서는 `list-assessment-runs`을 사용하는 방법을 보여 줍니다.

AWS CLI

평가 실행을 나열하려면

다음 `list-assessment-runs` 명령은 기존 평가 실행을 모두 나열합니다.

```
aws inspector list-assessment-runs
```

출력:

```
{
    "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"
    ]
}
```

자세한 내용은 Amazon Inspector 사용 설명서의 Amazon Inspector 평가 템플릿 및 평가 실행을 참조하세요. Amazon Inspector

- 자세한 API 내용은 명령 참조ListAssessmentRuns의 섹션을 참조하세요. AWS CLI

### `list-assessment-targets`

다음 코드 예시에서는 `list-assessment-targets`을 사용하는 방법을 보여 줍니다.

AWS CLI

평가 대상을 나열하려면

다음 `list-assessment-targets` 명령은 기존 평가 대상을 모두 나열합니다.

```
aws inspector list-assessment-targets
```

출력:

```
{
       "assessmentTargetArns": [
       "arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq"
       ]
}
```

자세한 내용은 Amazon Inspector 가이드의 Amazon Inspector 평가 대상을 참조하세요.

- 자세한 API 내용은 명령 참조ListAssessmentTargets의 섹션을 참조하세요. AWS CLI

### `list-assessment-templates`

다음 코드 예시에서는 `list-assessment-templates`을 사용하는 방법을 보여 줍니다.

AWS CLI

평가 템플릿을 나열하려면

다음 `list-assessment-templates` 명령은 기존 평가 템플릿을 모두 나열합니다.

```
aws inspector list-assessment-templates
```

출력:

```
{
       "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"
       ]
}
```

자세한 내용은 Amazon Inspector 가이드의 Amazon Inspector 평가 템플릿 및 평가 실행을 참조하세요.

- 자세한 API 내용은 명령 참조ListAssessmentTemplates의 섹션을 참조하세요. AWS CLI

### list-coverage-statistics

다음 코드 예시에서는 `list-coverage-statistics`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 그룹별로 적용 범위 통계를 나열하려면

다음 `list-coverage-statistics` 예제에서는 AWS 환경의 적용 범위 통계를 그룹별로 나열합니다.

```
aws inspector2 list-coverage-statistics \
   --group-by RESOURCE_TYPE
```

출력:

```
{
    "countsByGroup": [
        {
            "count": 56,
            "groupKey": "AWS_LAMBDA_FUNCTION"
        },
        {
            "count": 27,
```

```
            "groupKey": "AWS_ECR_REPOSITORY"
        },
        {
            "count": 18,
            "groupKey": "AWS_EC2_INSTANCE"
        },
        {
            "count": 3,
            "groupKey": "AWS_ECR_CONTAINER_IMAGE"
        },
        {
            "count": 1,
            "groupKey": "AWS_ACCOUNT"
        }
    ],
    "totalCounts": 105
}
```

자세한 내용은 Amazon Inspector 사용 설명서의 AWS 환경의 Amazon Inspector 적용 범위 평가를 참조하세요. Amazon Inspector

예제 2: 리소스 유형별로 적용 범위 통계를 나열하려면

다음 `list-coverage-statistics` 예제에서는 리소스 유형별로 AWS 환경의 적용 범위 통계를 나열합니다.

```
aws inspector2 list-coverage-statistics
    --filter-criteria '{"resourceType":
[{"comparison":"EQUALS","value":"AWS_ECR_REPOSITORY"}]}'
    --group-by SCAN_STATUS_REASON
```

출력:

```
{
    "countsByGroup": [
        {
            "count": 27,
            "groupKey": "SUCCESSFUL"
        }
    ],
    "totalCounts": 27
}
```

자세한 내용은 Amazon Inspector 사용 설명서의 AWS 환경의 Amazon Inspector 적용 범위 평가를 참조하세요. Amazon Inspector

예제 3: ECR리포지토리 이름별로 적용 범위 통계를 나열하려면

다음 `list-coverage-statistics` 예제에서는 ECR리포지토리 이름별로 AWS 환경의 적용 범위 통계를 나열합니다.

```
aws inspector2 list-coverage-statistics
   --filter-criteria '{"ecrRepositoryName":
[{"comparison":"EQUALS","value":"debian"}]}'
   --group-by SCAN_STATUS_REASON
```

출력:

```
{
    "countsByGroup": [
        {
            "count": 3,
            "groupKey": "SUCCESSFUL"
        }
    ],
    "totalCounts": 3
}
```

자세한 내용은 Amazon Inspector 사용 설명서의 AWS 환경에 대한 Amazon Inspector 적용 범위 평가를 참조하세요. Amazon Inspector

- 자세한 API 내용은 명령 참조ListCoverageStatistics의 섹션을 참조하세요. AWS CLI

### `list-coverage`

다음 코드 예시에서는 `list-coverage`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 환경에 대한 적용 범위 세부 정보를 나열하려면

다음 `list-coverage` 예제에서는 환경의 적용 범위 세부 정보를 나열합니다.

```
aws inspector2 list-coverage
```

출력:

```
{
    "coveredResources": [
        {
            "accountId": "123456789012",
            "lastScannedAt": "2024-05-20T16:23:20-07:00",
            "resourceId": "i-EXAMPLE55555555555",
            "resourceMetadata": {
                "ec2": {
                    "amiId": "ami-EXAMPLE6666666666",
                    "platform": "LINUX"
                }
            },
            "resourceType": "AWS_EC2_INSTANCE",
            "scanStatus": {
                "reason": "SUCCESSFUL",
                "statusCode": "ACTIVE"
            },
            "scanType": "PACKAGE"
        }
    ]
}
```

예제 2: Lambda 함수 리소스 유형에 대한 적용 범위 세부 정보를 나열하려면

다음 `list-coverage` 예제에서는 Lamda 함수 리소스 유형 세부 정보를 나열합니다.

```
aws inspector2 list-coverage
    --filter-criteria '{"resourceType":
[{"comparison":"EQUALS","value":"AWS_LAMBDA_FUNCTION"}]}'
```

출력:

```
{
    "coveredResources": [
        {
            "accountId": "123456789012",
            "resourceId": "arn:aws:lambda:us-west-2:123456789012:function:Eval-
container-scan-results:$LATEST",
            "resourceMetadata": {
                "lambdaFunction": {
```

```
                    "functionName": "Eval-container-scan-results",
                    "functionTags": {},
                    "layers": [],
                    "runtime": "PYTHON_3_7"
                }
            },
            "resourceType": "AWS_LAMBDA_FUNCTION",
            "scanStatus": {
                "reason": "SUCCESSFUL",
                "statusCode": "ACTIVE"
            },
            "scanType": "CODE"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListCoverage의 섹션을 참조하세요. AWS CLI

### `list-delegated-admin-accounts`

다음 코드 예시에서는 `list-delegated-admin-accounts`을 사용하는 방법을 보여 줍니다.

AWS CLI

조직의 위임된 관리자 계정에 대한 정보를 나열하려면

다음 `list-delegated-admin-accounts` 예제에서는 조직의 위임된 관리자 계정에 대한 정보를 나열합니다.

```
aws inspector2 list-delegated-admin-accounts
```

출력:

```
{
    "delegatedAdminAccounts": [
        {
            "accountId": "123456789012",
            "status": "ENABLED"
        }
    ]
}
```

자세한 내용은 Amazon Inspector 사용 설명서의 Amazon Inspector에 위임된 관리자 지정을 참조하세요. Amazon Inspector

- 자세한 API 내용은 명령 참조ListDelegatedAdminAccounts의 섹션을 참조하세요. AWS CLI

### `list-event-subscriptions`

다음 코드 예시에서는 `list-event-subscriptions`을 사용하는 방법을 보여 줍니다.

AWS CLI

이벤트 구독을 나열하려면

다음 `list-event-subscriptions` 명령은 의 ARN 를 사용하여 평가 템플릿에 대한 모든 이벤트 구독을 나열합니다`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
```

출력:

```
{
      "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"
        }
      ]
}
```

자세한 내용은 Amazon Inspector 가이드의 Amazon Inspector 평가 템플릿 및 평가 실행을 참조하세요.

- 자세한 API 내용은 명령 참조ListEventSubscriptions의 섹션을 참조하세요. AWS CLI

### list-filters

다음 코드 예시에서는 list-filters을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon Inspector를 활성화하는 데 사용한 계정과 연결된 필터를 나열하려면

다음 list-filters 예제에서는 Amazon Inspector를 활성화하는 데 사용한 계정과 연결된 필터를 나열합니다.

```
aws inspector2 list-filters
```

출력:

```
{
    "filters": [
        {
            "action": "SUPPRESS",
            "arn": "arn:aws:inspector2:us-west-2:123456789012:owner/o-EXAMPLE222/
filter/EXAMPLE444444444",
            "createdAt": "2024-05-15T21:11:08.602000+00:00",
            "criteria": {
                "resourceType": [
                    {
                        "comparison": "EQUALS",
                        "value": "AWS_EC2_INSTANCE"
                    },
                ]
            },
            "description": "This suppression rule omits EC2 instance type findings",
            "name": "ExampleSuppressionRuleEC2",
            "ownerId": "o-EXAMPLE222",
            "tags": {},
            "updatedAt": "2024-05-15T21:11:08.602000+00:00"
        },
        {
            "action": "SUPPRESS",
            "arn": "arn:aws:inspector2:us-east-1:813737243517:owner/o-EXAMPLE222/
filter/EXAMPLE444444444",
            "createdAt": "2024-05-15T21:28:27.054000+00:00",
            "criteria": {
                "resourceType": [
```

```
                    {
                        "comparison": "EQUALS",
                        "value": "AWS_ECR_INSTANCE"
                    }
                ]
            },
            "description": "This suppression rule omits ECR instance type findings",
            "name": "ExampleSuppressionRuleECR",
            "ownerId": "o-EXAMPLE222",
            "tags": {},
            "updatedAt": "2024-05-15T21:28:27.054000+00:00"
        }
    ]
}
```

자세한 내용은 Amazon Inspector 사용 설명서의 Amazon Inspector 결과 필터링을 참조하세요. Amazon Inspector

- 자세한 API 내용은 명령 참조ListFilters의 섹션을 참조하세요. AWS CLI

### `list-findings`

다음 코드 예시에서는 `list-findings`을 사용하는 방법을 보여 줍니다.

AWS CLI

조사 결과를 나열하려면

다음 `list-findings` 명령은 생성된 모든 조사 결과를 나열합니다.

```
aws inspector list-findings
```

출력:

```
{
        "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"
      ]
}
```

자세한 내용은 Amazon Inspector 안내서의 Amazon Inspector 조사 결과를 참조하세요.

- 자세한 API 내용은 명령 참조ListFindings의 섹션을 참조하세요. AWS CLI

### list-rules-packages

다음 코드 예시에서는 list-rules-packages을 사용하는 방법을 보여 줍니다.

AWS CLI

규칙 패키지를 나열하려면

다음 list-rules-packages 명령은 사용 가능한 모든 Inspector 규칙 패키지를 나열합니다.

```
aws inspector list-rules-packages
```

출력:

```
{
       "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-JJOtZiqQ",
         "arn:aws:inspector:us-west-2:758058086616:rulespackage/0-vg5GGHSD"
       ]
 }
```

자세한 내용은 Amazon Inspector 가이드의 Amazon Inspector 규칙 패키지 및 규칙을 참조하세요.

- 자세한 API 내용은 명령 참조ListRulesPackages의 섹션을 참조하세요. AWS CLI

### list-tags-for-resource

다음 코드 예시에서는 list-tags-for-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 대한 태그를 나열하려면

다음 list-tags-for-resource 명령은 ARN의 를 사용하여 평가 템플릿과 연결된 모든 태그를 나열합니다arn: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
```

출력:

```
{
      "tags": [
        {
              "key": "Name",
              "value": "Example"
        }
      ]
}
```

자세한 내용은 Amazon Inspector 가이드의 Amazon Inspector 평가 템플릿 및 평가 실행을 참조하세요.

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### `list-usage-totals`

다음 코드 예시에서는 `list-usage-totals`을 사용하는 방법을 보여 줍니다.

AWS CLI

지난 30일 동안의 사용량 합계를 나열하려면

다음 `list-usage-totals` 예제에서는 지난 30일 동안의 사용량 합계를 나열합니다.

```
aws inspector2 list-usage-totals
```

출력:

```
{
    "totals": [
        {
            "accountId": "123456789012",
            "usage": [
                {
                    "currency": "USD",
```

```
                    "estimatedMonthlyCost": 4.6022044647,
                    "total": 1893.4784083333334,
                    "type": "EC2_AGENTLESS_INSTANCE_HOURS"
                },
                {
                    "currency": "USD",
                    "estimatedMonthlyCost": 18.892449279,
                    "total": 10882.050784722222,
                    "type": "EC2_INSTANCE_HOURS"
                },
                {
                    "currency": "USD",
                    "estimatedMonthlyCost": 5.4525363736,
                    "total": 6543.043648333333,
                    "type": "LAMBDA_FUNCTION_CODE_HOURS"
                },
                {
                    "currency": "USD",
                    "estimatedMonthlyCost": 3.9064080309,
                    "total": 9375.379274166668,
                    "type": "LAMBDA_FUNCTION_HOURS"
                },
                {
                    "currency": "USD",
                    "estimatedMonthlyCost": 0.06,
                    "total": 6.0,
                    "type": "ECR_RESCAN"
                },
                {
                    "currency": "USD",
                    "estimatedMonthlyCost": 0.09,
                    "total": 1.0,
                    "type": "ECR_INITIAL_SCAN"
                }
            ]
        }
    ]
}
```


자세한 내용은 Amazon Inspector 사용 설명서의 Amazon Inspector에서 사용량 및 비용 모니터링을 참조하세요. Amazon Inspector

- 자세한 API 내용은 명령 참조ListUsageTotals의 섹션을 참조하세요. AWS CLI

### preview-agents

다음 코드 예시에서는 preview-agents을 사용하는 방법을 보여 줍니다.

AWS CLI

에이전트를 미리 보려면

다음 preview-agents 명령은 ARN 의 를 사용하여 평가 대상의 일부인 EC2 인스턴스에 설치된 에이전트를 미리 봅니다arn: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
```

출력:

```
{
      "agentPreviews": [
        {
              "agentId": "i-49113b93"
        }
      ]
}
```

자세한 내용은 Amazon Inspector 가이드의 Amazon Inspector 평가 대상을 참조하세요.

- 자세한 API 내용은 명령 참조PreviewAgents의 섹션을 참조하세요. AWS CLI

### register-cross-account-access-role

다음 코드 예시에서는 register-cross-account-access-role을 사용하는 방법을 보여 줍니다.

AWS CLI

교차 계정 액세스 역할을 등록하려면

다음 register-cross-account-access-role 명령은 Preview-agents 명령을 호출할 때 평가 실행 시작 시 Amazon InspectorARNarn:aws:iam::123456789012:role/inspector가 EC2 인스턴스를 나열하는 데 사용하는 의 에 IAM 역할을 등록합니다.

```
aws inspector register-cross-account-access-role --role-
arn arn:aws:iam::123456789012:role/inspector
```

자세한 내용은 Amazon Inspector 안내서의 Amazon Inspector 설정을 참조하세요.

- 자세한 API 내용은 명령 참조RegisterCrossAccountAccessRole의 섹션을 참조하세요. AWS CLI

### remove-attributes-from-findings

다음 코드 예시에서는 remove-attributes-from-findings을 사용하는 방법을 보여 줍니다.

AWS CLI

조사 결과에서 속성을 제거하려면

다음 remove-attributes-from-finding 명령은 의 키를 사용하여 속성을 제거하고 ARN 의 키를 사용하여 결과example에서 Example 값을 제거합니다arn:aws:inspector:us-west-2:123456789012:target/0-0kFIPusq/template/0-8l1VIE0D/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/0-8l1VIE0D/run/0-Z02cjjug/finding/0-
T8yM9mEU --attribute-keys key=Example,value=example
```

출력:

```
{
      "failedItems": {}
}
```

자세한 내용은 Amazon Inspector 안내서의 Amazon Inspector 조사 결과를 참조하세요.

- 자세한 API 내용은 명령 참조RemoveAttributesFromFindings의 섹션을 참조하세요. AWS CLI

### set-tags-for-resource

다음 코드 예시에서는 set-tags-for-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 대한 태그를 설정하려면

다음 set-tags-for-resource 명령은 키가 인 태그Example와 값이 인 태그를 ARN인 example 평가 템플릿에 설정합니다arn: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
```

자세한 내용은 Amazon Inspector 가이드의 Amazon Inspector 평가 템플릿 및 평가 실행을 참조하세요.

- 자세한 API 내용은 명령 참조SetTagsForResource의 섹션을 참조하세요. AWS CLI

### start-assessment-run

다음 코드 예시에서는 start-assessment-run을 사용하는 방법을 보여 줍니다.

AWS CLI

평가 실행을 시작하려면

다음 start-assessment-run 명령은 ARN 의 를 사용하여 평가 템플릿을 examplerun 사용하여 라는 평가 실행을 시작합니다arn: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
```

출력:

```
{
      "assessmentRunArn": "arn:aws:inspector:us-west-2:123456789012:target/0-
nvgVhaxX/template/0-it5r2S4T/run/0-jOoroxyY"
}
```

자세한 내용은 Amazon Inspector 가이드의 Amazon Inspector 평가 템플릿 및 평가 실행을 참조하세요.

- 자세한 API 내용은 명령 참조StartAssessmentRun의 섹션을 참조하세요. AWS CLI

### stop-assessment-run

다음 코드 예시에서는 stop-assessment-run을 사용하는 방법을 보여 줍니다.

AWS CLI

평가 실행을 중지하려면

다음 stop-assessment-run 명령은 ARN 의 를 사용하여 평가 실행을 중지합니다arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T/run/0-jOoroxyY.

```
aws inspector stop-assessment-run --assessment-run-arn arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX/template/0-it5r2S4T/run/0-jOoroxyY
```

자세한 내용은 Amazon Inspector 가이드의 Amazon Inspector 평가 템플릿 및 평가 실행을 참조하세요.

- 자세한 API 내용은 명령 참조StopAssessmentRun의 섹션을 참조하세요. AWS CLI

### subscribe-to-event

다음 코드 예시에서는 subscribe-to-event을 사용하는 방법을 보여 줍니다.

AWS CLI

이벤트를 구독하려면

다음 예제에서는 의 를 사용하여 ASSESSMENT_RUN_COMPLETED 이벤트에 대한 Amazon SNS 알림을 주제로 보내는 프로세스를 활성화합니다. ARN 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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Inspector 가이드의 Amazon Inspector 평가 템플릿 및 평가 실행을 참조하세요. Amazon Inspector

- 자세한 API 내용은 명령 참조SubscribeToEvent의 섹션을 참조하세요. AWS CLI

### unsubscribe-from-event

다음 코드 예시에서는 unsubscribe-from-event을 사용하는 방법을 보여 줍니다.

AWS CLI

이벤트 구독을 취소하려면

다음 unsubscribe-from-event 명령은 ARN 의 를 사용하여 ASSESSMENT_RUN_COMPLETED 이벤트에 대한 Amazon SNS 알림을 주제로 보내는 프로세스를 비활성화합니다arn: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
```

자세한 내용은 Amazon Inspector 가이드의 Amazon Inspector 평가 템플릿 및 평가 실행을 참조하세요.

- 자세한 API 내용은 명령 참조UnsubscribeFromEvent의 섹션을 참조하세요. AWS CLI

### update-assessment-target

다음 코드 예시에서는 update-assessment-target을 사용하는 방법을 보여 줍니다.

AWS CLI

평가 대상을 업데이트하려면

다음 update-assessment-target 명령은 평가 대상을 ARN 의 arn:aws:inspector:us-west-2:123456789012:target/0-nvgVhaxX 와 이름으로 Example업데이트하고 리소스 그룹을 ARN의 로 업데이트합니다arn: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
```

자세한 내용은 Amazon Inspector 가이드의 Amazon Inspector 평가 대상을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateAssessmentTarget의 섹션을 참조하세요. AWS CLI

### update-filter

다음 코드 예시에서는 update-filter을 사용하는 방법을 보여 줍니다.

AWS CLI

필터를 업데이트하려면

다음 update-filter 예제에서는 ECR 인스턴스 조사 결과 대신 Lambda 조사 결과를 생략하도록 필터를 업데이트합니다.

```
aws inspector2 update-filter \
   --filter-arn "arn:aws:inspector2:us-west-2:123456789012:owner/o-EXAMPLE222/
filter/EXAMPLE444444444" \
   --name "ExampleSuppressionRuleLambda" \
   --description "This suppression rule omits Lambda instance findings" \
   --reason "Updating filter to omit Lambda instance findings instead of ECR
 instance findings"
```

출력:

```
{
    "filters": [
        {
            "action": "SUPPRESS",
            "arn": "arn:aws:inspector2:us-west-2:123456789012:owner/o-EXAMPLE222/
filter/EXAMPLE444444444",
            "createdAt": "2024-05-15T21:28:27.054000+00:00",
            "criteria": {
                "resourceType": [
                    {
                        "comparison": "EQUALS",
                        "value": "AWS_ECR_INSTANCE"
                    }
                ]
            },
            "description": "This suppression rule omits Lambda instance findings",
            "name": "ExampleSuppressionRuleLambda",
            "ownerId": "o-EXAMPLE222",
```

```
            "reason": "Updating filter to omit Lambda instance findings instead of
 ECR instance findings",
            "tags": {},
            "updatedAt": "2024-05-15T22:23:13.665000+00:00"
        }
    ]
}
```

자세한 내용은 Amazon Inspector 사용 설명서의 Amazon Inspector에서 결과 관리를 참조하세요. Amazon Inspector

- 자세한 API 내용은 명령 참조UpdateFilter의 섹션을 참조하세요. AWS CLI

# AWS IoT 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS IoT.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### `accept-certificate-transfer`

다음 코드 예시에서는 `accept-certificate-transfer`을 사용하는 방법을 보여 줍니다.

AWS CLI

다른 AWS 계정에서 전송된 디바이스 인증서를 수락하려면

다음 `accept-certificate-transfer` 예제에서는 다른 AWS 계정에서 전송된 디바이스 인증서를 수락합니다. 인증서는 ID로 식별됩니다.

```
aws iot accept-certificate-transfer \
    --certificate-
id 488b6a7f2acdeb00a77384e63c4e40b18bEXAMPLEe57b7272ba44c45e3448142
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT Core 개발자 안내서의 다른 계정으로 인증서 전송을 참조하세요.

- 자세한 API 내용은 명령 참조AcceptCertificateTransfer의 섹션을 참조하세요. AWS CLI

### add-thing-to-billing-group

다음 코드 예시에서는 add-thing-to-billing-group을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: 결제 그룹에 이름으로 사물을 추가하려면

다음 add-thing-to-billing-group 예에서는 라는 결제 그룹에 라는 사물MyLightBulb을 추가합니다GroupOne.

```
aws iot add-thing-to-billing-group \
    --billing-group-name GroupOne \
    --thing-name MyLightBulb
```

이 명령은 출력을 생성하지 않습니다.

예제 2: 결제 그룹에 로 사물ARN을 추가하려면

다음 add-thing-to-billing-group 예에서는 지정된 가 있는 사물을 지정된 가 있는 ARN 결제 그룹에 추가합니다ARN. 여러 AWS 리전 또는 계정을 사용하는 경우 를 지정하는 ARN 것이 유용합니다. 올바른 리전 및 계정에 를 추가하는 데 도움이 될 수 있습니다.

```
aws iot add-thing-to-thing-group \
    --billing-group-arn "arn:aws:iot:us-west-2:123456789012:billinggroup/GroupOne" \
    --thing-arn "arn:aws:iot:us-west-2:123456789012:thing/MyOtherLightBulb"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IoT 개발자 안내서의 결제 그룹을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조AddThingToBillingGroup의 섹션을 참조하세요. AWS CLI

### add-thing-to-thing-group

다음 코드 예시에서는 add-thing-to-thing-group을 사용하는 방법을 보여 줍니다.

AWS CLI

그룹에 사물을 추가하려면

다음 add-thing-to-thing-group 예제에서는 지정된 사물을 지정된 사물 그룹에 추가합니다.

```
aws iot add-thing-to-thing-group \
    --thing-name MyLightBulb \
    --thing-group-name LightBulbs
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IoT 개발자 안내서의 사물 그룹을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조AddThingToThingGroup의 섹션을 참조하세요. AWS CLI

### associate-targets-with-job

다음 코드 예시에서는 associate-targets-with-job을 사용하는 방법을 보여 줍니다.

AWS CLI

사물 그룹을 연속 작업과 연결하려면

다음 associate-targets-with-job 예제에서는 지정된 사물 그룹을 지정된 연속 작업과 연결합니다.

```
aws iot associate-targets-with-job \
    --targets "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs" \
    --job-id "example-job-04"
```

출력:

```
{
    "jobArn": "arn:aws:iot:us-west-2:123456789012:job/example-job-04",
```

```
    "jobId": "example-job-04",
    "description": "example continuous job"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 작업 생성 및 관리(CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조AssociateTargetsWithJob의 섹션을 참조하세요. AWS CLI

### attach-policy

다음 코드 예시에서는 attach-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 사물 그룹에 정책을 연결하려면

다음 attach-policy 예제에서는 지정된 정책을 해당 로 식별된 사물 그룹에 연결합니다ARN.

```
aws iot attach-policy \
    --target "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs" \
    --policy-name "UpdateDeviceCertPolicy"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IoT 개발자 안내서의 사물 그룹을 참조하세요. AWS IoT

예제 2: 인증서를 정책으로 연결하려면

다음 attach-policy 예제에서는 인증서를 통해 지정된 보안 주체UpdateDeviceCertPolicy에 정책을 연결합니다.

```
aws iot attach-policy \
    --policy-name UpdateDeviceCertPolicy \
    --target "arn:aws:iot:us-
west-2:123456789012:cert/4f0ba725787aa94d67d2fca420eca022242532e8b3c58e7465c7778b443fd65e"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 디바이스 인증서에 IoT 정책 연결을 AWS 참조하세요 IoT.

- 자세한 API 내용은 명령 참조AttachPolicy의 섹션을 참조하세요. AWS CLI

### attach-security-profile

다음 코드 예시에서는 attach-security-profile을 사용하는 방법을 보여 줍니다.

AWS CLI

보안 프로파일을 등록되지 않은 모든 디바이스와 연결하려면

다음 attach-security-profile 예제는 라는 AWS IoT Device Defender 보안 프로파일을 이 AWS 계정의 us-west-2 리전에 있는 모든 미등록 디바이스Testprofile와 연결합니다.

```
aws iot attach-security-profile \
    --security-profile-name Testprofile \
    --security-profile-target-arn "arn:aws:iot:us-west-2:123456789012:all/
unregistered-things"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 Detect Commands를 참조하세요.

- 자세한 API 내용은 명령 참조AttachSecurityProfile의 섹션을 참조하세요. AWS CLI

### attach-thing-principal

다음 코드 예시에서는 attach-thing-principal을 사용하는 방법을 보여 줍니다.

AWS CLI

사물에 인증서를 연결하려면

다음 attach-thing-principal 예제에서는 인증서를 MyTemperatureSensor 사물에 연결합니다. 인증서는 로 식별됩니다ARN. AWS IoT 콘솔에서 인증서에 ARN 대한 를 찾을 수 있습니다.

```
aws iot attach-thing-principal \
    --thing-name MyTemperatureSensor \
    --principal arn:aws:iot:us-
west-2:123456789012:cert/2e1eb273792174ec2b9bf4e9b37e6c6c692345499506002a35159767055278e8
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 레지스트리를 사용하여 사물을 관리하는 방법을 참조하세요.

- 자세한 API 내용은 명령 참조AttachThingPrincipal의 섹션을 참조하세요. AWS CLI

### cancel-audit-mitigation-actions-task

다음 코드 예시에서는 cancel-audit-mitigation-actions-task을 사용하는 방법을 보여 줍니다.

AWS CLI

감사 완화 작업 작업을 취소하려면

다음 cancel-audit-mitigations-action-task 예제에서는 지정된 작업에 대한 완화 작업 적용을 취소합니다. 이미 완료된 작업은 취소할 수 없습니다.

```
aws iot cancel-audit-mitigation-actions-task
    --task-id "myActionsTaskId"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 CancelAuditMitigationActionsTask (완화 작업 명령)을 참조하세요.

- 자세한 API 내용은 명령 참조CancelAuditMitigationActionsTask의 섹션을 참조하세요. AWS CLI

### cancel-audit-task

다음 코드 예시에서는 cancel-audit-task을 사용하는 방법을 보여 줍니다.

AWS CLI

감사 작업을 취소하려면

다음 cancel-audit-task 예제에서는 지정된 작업 ID로 감사 작업을 취소합니다. 완료된 태스크는 취소할 수 없습니다.

```
aws iot cancel-audit-task \
    --task-id a3aea009955e501a31b764abe1bebd3d
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 감사 명령을 참조하세요.

- 자세한 API 내용은 명령 참조CancelAuditTask의 섹션을 참조하세요. AWS CLI

### cancel-certificate-transfer

다음 코드 예시에서는 cancel-certificate-transfer을 사용하는 방법을 보여 줍니다.

AWS CLI

다른 AWS 계정으로 인증서 전송을 취소하려면

다음 cancel-certificate-transfer 예제에서는 지정된 인증서 전송의 전송을 취소합니다. 인증서는 인증서 ID로 식별됩니다. AWS IoT 콘솔에서 인증서의 ID를 찾을 수 있습니다.

```
aws iot cancel-certificate-transfer \
    --certificate-
id f0f33678c7c9a046e5cc87b2b1a58dfa0beec26db78addd5e605d630e05c7fc8
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT Core 개발자 안내서의 다른 계정으로 인증서 전송을 참조하세요.

- 자세한 API 내용은 명령 참조CancelCertificateTransfer의 섹션을 참조하세요. AWS CLI

### cancel-job-execution

다음 코드 예시에서는 cancel-job-execution을 사용하는 방법을 보여 줍니다.

AWS CLI

디바이스에서 작업 실행을 취소하려면

다음 cancel-job-execution 예제에서는 디바이스에서 지정된 작업의 실행을 취소합니다. 작업이 QUEUED 상태가 아닌 경우 --force 파라미터를 추가해야 합니다.

```
aws iot cancel-job-execution \
    --job-id "example-job-03" \
    --thing-name "MyRPi"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 작업 생성 및 관리(CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조CancelJobExecution의 섹션을 참조하세요. AWS CLI

### cancel-job

다음 코드 예시에서는 cancel-job을 사용하는 방법을 보여 줍니다.

AWS CLI

작업을 취소하려면

다음 cancel-job 예제에서는 지정된 작업을 취소합니다.

```
aws iot cancel-job \
    --job-job "example-job-03"
```

출력:

```
{
    "jobArn": "arn:aws:iot:us-west-2:123456789012:job/example-job-03",
    "jobId": "example-job-03",
    "description": "example job test"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 작업 생성 및 관리(CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조CancelJob의 섹션을 참조하세요. AWS CLI

### clear-default-authorizer

다음 코드 예시에서는 clear-default-authorizer을 사용하는 방법을 보여 줍니다.

AWS CLI

기본 권한 부여자를 지우려면

다음 clear-default-authorizer 예제에서는 현재 구성된 기본 사용자 지정 권한 부여자를 지웁니다. 이 명령을 실행한 후에는 기본 권한 부여자가 없습니다. 사용자 지정 권한 부여자를 사용하는 경우 HTTP 요청 헤더에서 이름으로 지정해야 합니다.

```
aws iot clear-default-authorizer
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IoT 참조ClearDefaultAuthorizer의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조ClearDefaultAuthorizer의 섹션을 참조하세요. AWS CLI

### `confirm-topic-rule-destination`

다음 코드 예시에서는 `confirm-topic-rule-destination`을 사용하는 방법을 보여 줍니다.

AWS CLI

주제 규칙 대상을 확인하려면

다음 `confirm-topic-rule-destination` 예제에서는 HTTP 엔드포인트에서 수신된 확인 토큰을 사용하여 주제 규칙 대상을 확인합니다.

```
aws iot confirm-topic-rule-destination \
    --confirmation-token "AYADeIcmtq-
ZkxfpiWIQqHWM5ucAXwABABVhd3MtY3J5cHRvLXB1YmxpYy1rZXkAREFxY1E0UmlGeDg0V21BZWZ1VjZtZWFRVUJJUkt
aywpPqg8YEsa1lD4B40aJ2s1wEHKMybiF1RoOZzYisI0IvslzQY5UmCkqq3tV-3f7-
nKfosgIAAAADAAAEAAAAAAAAAAAAAAAAAAi9RMgy-
V19V9m6Iw2xfbw_____wAAAAEAAAAAAAAAAAAAAAEAAAB1hw4SokgUcxiJ3gTO6n50NLJVpzyQR1UmPIj5sShqXEQGc0
iufgrzTePl8RZYOWr0O6Aj9DiVzJZx-1iD6Pu-
G6PUw1kaO7Knzs2B4AD0qfrHUF4pYRTvyUgBnMGUCMQC8ZRmhKqntd_c6Kgrow3bMUDBvNqo2qZr8Z8Jm2rzgseROlAn
PIetJ803Z4IlIlF8xXlcdPGP-PV1dOXFemyL8g"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 주제 규칙 대상 확인을 참조하세요.

- 자세한 API 내용은 명령 참조ConfirmTopicRuleDestination의 섹션을 참조하세요. AWS CLI

### `create-audit-suppression`

다음 코드 예시에서는 `create-audit-suppression`을 사용하는 방법을 보여 줍니다.

AWS CLI

감사 결과 금지를 생성하려면

다음 `create-audit-suppression` 예제에서는 과도하게 허용되었다는 플래그가 지정된 “virtualMachinePolicy”라는 정책에 대한 감사 결과 금지를 생성합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IoT 개발자 안내서의 감사 결과 금지를 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조CreateAuditSuppression의 섹션을 참조하세요. AWS CLI

### `create-authorizer`

다음 코드 예시에서는 `create-authorizer`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 권한 부여자를 생성하려면

다음 `create-authorizer` 예제에서는 지정된 Lambda 함수를 사용자 지정 인증 서비스의 일부로 사용하는 사용자 지정 권한 부여자를 생성합니다.

```
   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+7ENxAio9q6QD4FFqjGZsvjtYwjoe1RKK0U8Eq9xb5O3kRSmyIwTzwzm/f4Gf0Y
ZUloJ+t3PUUwHrmbYTAgTrCUgRFygjfgVwGCPs5ZAX4Eyqt5cr+AIHIiUDbxSa7p
zwOBKPeic0asNJpqT8PkBbRaKyleJh5oo81NDHHmVtbBm5A5YiJjqYXLaVAowKzZ
+GqsNvAQ9Jy1wI2VrEa1OfL8flDB/BJLm7zjpfPOHDJQgID0XnZwAlNnZcOhCwIx
50g2LW2Oy9R/dmqtDmJiVP97Z4GykxPvwlYHrUXY0iW1R3AR/Ac1NhCTGZMwVDB1
lQIDAQAB
-----END PUBLIC KEY-----"
```

출력:

```
{
    "authorizerName": "CustomAuthorizer",
    "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/
CustomAuthorizer2"
}
```

자세한 내용은 IoT 참조CreateAuthorizer의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조CreateAuthorizer의 섹션을 참조하세요. AWS CLI

### `create-billing-group`

다음 코드 예시에서는 `create-billing-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

결제 그룹을 생성하려면

다음 `create-billing-group` 예제에서는 라는 간단한 결제 그룹을 생성합니다`GroupOne`.

```
aws iot create-billing-group \
    --billing-group-name GroupOne
```

출력:

```
{
    "billingGroupName": "GroupOne",
    "billingGroupArn": "arn:aws:iot:us-west-2:123456789012:billinggroup/GroupOne",
    "billingGroupId": "103de383-114b-4f51-8266-18f209ef5562"
}
```

자세한 내용은 IoT 개발자 안내서의 결제 그룹을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조CreateBillingGroup의 섹션을 참조하세요. AWS CLI

### `create-certificate-from-csr`

다음 코드 예시에서는 `create-certificate-from-csr`을 사용하는 방법을 보여 줍니다.

AWS CLI

인증서 서명 요청에서 디바이스 인증서를 생성하려면(CSR)

다음 create-certificate-from-csr 예제에서는 에서 디바이스 인증서를 생성합니다CSR. openssl 명령을 사용하여 를 생성할 수 있습니다CSR.

```
aws iot create-certificate-from-csr \
    --certificate-signing-request=file://certificate.csr
```

출력:

```
{
    "certificateArn": "arn:aws:iot:us-west-2:123456789012:cert/
c0c57bbc8baaf4631a9a0345c957657f5e710473e3ddbee1428d216d54d53ac9",
        "certificateId":
 "c0c57bbc8baaf4631a9a0345c957657f5e710473e3ddbee1428d216d54d53ac9",
        "certificatePem": "<certificate-text>"
}
```

자세한 내용은 IoT 참조CreateCertificateFromCSR의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조CreateCertificateFromCsr의 섹션을 참조하세요. AWS CLI

### create-custom-metric

다음 코드 예시에서는 create-custom-metric을 사용하는 방법을 보여 줍니다.

AWS CLI

디바이스가 Device Defender에 게시한 사용자 지정 지표를 생성하려면

다음 create-custom-metric 예제에서는 배터리 비율을 측정하는 사용자 지정 지표를 생성합니다.

```
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"
```

출력:

```
{
    "metricName": "batteryPercentage",
```

```
    "metricArn": "arn:aws:iot:us-east-1:1234564789012:custommetric/
batteryPercentage"
}
```

자세한 내용은 AWS IoT Core 개발자 안내서의 사용자 지정 지표를 참조하세요.

- 자세한 API 내용은 명령 참조CreateCustomMetric의 섹션을 참조하세요. AWS CLI

### `create-dimension`

다음 코드 예시에서는 `create-dimension`을 사용하는 방법을 보여 줍니다.

AWS CLI

차원을 생성하려면

다음은 라는 단일 주제 필터를 사용하여 차원을 `create-dimension` 생성합니다`TopicFilterForAuthMessages`.

```
aws iot create-dimension \
    --name TopicFilterForAuthMessages \
    --type TOPIC_FILTER \
    --string-values device/+/auth
```

출력:

```
{
    "name": "TopicFilterForAuthMessages",
    "arn": "arn:aws:iot:eu-west-2:123456789012:dimension/TopicFilterForAuthMessages"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 Detect Commands를 참조하세요.

- 자세한 API 내용은 명령 참조CreateDimension의 섹션을 참조하세요. AWS CLI

### `create-domain-configuration`

다음 코드 예시에서는 `create-domain-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

도메인 구성을 생성하려면

다음 create-domain-configuration 예제에서는 서비스 유형이 인 AWS관리형 도메인 구성을 생성합니다DATA.

```
aws iot create-domain-configuration \
    --domain-configuration-name "additionalDataDomain" \
    --service-type "DATA"
```

출력:

```
{
    "domainConfigurationName": "additionalDataDomain",
    "domainConfigurationArn": "arn:aws:iot:us-
west-2:123456789012:domainconfiguration/additionalDataDomain/dikMh"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 구성 가능한 엔드포인트를 참조하세요.

- 자세한 API 내용은 명령 참조CreateDomainConfiguration의 섹션을 참조하세요. AWS CLI

### create-dynamic-thing-group

다음 코드 예시에서는 create-dynamic-thing-group을 사용하는 방법을 보여 줍니다.

AWS CLI

동적 사물 그룹을 생성하려면

다음 create-dynamic-thing-group 예제에서는 온도 속성이 60도보다 큰 사물이 포함된 동적 사물 그룹을 생성합니다. 동적 사물 그룹을 사용하려면 먼저 AWS IoT 플릿 인덱싱을 활성화해야 합니다.

```
aws iot create-dynamic-thing-group \
    --thing-group-name "RoomTooWarm" \
    --query-string "attributes.temperature>60"
```

출력:

```
{
    "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"
}
```

자세한 내용은 IoT 개발자 안내서의 동적 사물 그룹을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조CreateDynamicThingGroup의 섹션을 참조하세요. AWS CLI

### `create-job`

다음 코드 예시에서는 `create-job`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 작업 생성

다음 `create-job` 예제에서는 `MyRaspberryPi` 디바이스에 JSON 문서를 보내는 간단한 AWS IoT 작업을 생성합니다.

```
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
```

출력:

```
{
    "jobArn": "arn:aws:iot:us-west-2:123456789012:job/example-job-01",
    "jobId": "example-job-01",
    "description": "example job test"
}
```

예제 2: 연속 작업을 생성하려면

다음 `create-job` 예제에서는 대상으로 지정된 사물이 작업을 완료한 후 계속 실행되는 작업을 생성합니다. 이 예제에서 대상은 사물 그룹이므로 새 디바이스가 그룹에 추가되면 연속 작업이 이러한 새 사물에서 실행됩니다.

aws iot create-job --job-id “example-job-04” --targets “arn:aws:iot:us-west-2:123456789012:thinggroup/DeadBulbs" --document file://example-job.json --description “example continuous job” --target-selection CONTINUOUS

출력:

```
{
    "jobArn": "arn:aws:iot:us-west-2:123456789012:job/example-job-04",
    "jobId": "example-job-04",
    "description": "example continuous job"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 작업 생성 및 관리(CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조CreateJob의 섹션을 참조하세요. AWS CLI

### `create-keys-and-certificate`

다음 코드 예시에서는 `create-keys-and-certificate`을 사용하는 방법을 보여 줍니다.

AWS CLI

RSA 키 페어를 생성하고 X.509 인증서를 발급하려면

다음은 2048비트 RSA 키 페어를 `create-keys-and-certificate` 생성하고 발급된 퍼블릭 키를 사용하여 X.509 인증서를 발급합니다. AWS IoT가 이 인증서의 프라이빗 키를 제공하는 유일한 시간이므로 안전한 위치에 보관해야 합니다.

```
aws iot create-keys-and-certificate \
    --certificate-pem-outfile "myTest.cert.pem" \
    --public-key-outfile "myTest.public.key" \
    --private-key-outfile "myTest.private.key"
```

출력:

```
{
    "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/9894ba17925e663f1d29c23af4582b8e3b7619c31f3fbd93adcb51ae54b83dc2",
    "certificateId":
 "9894ba17925e663f1d29c23af4582b8e3b7619c31f3fbd93adcb51ae54b83dc2",
```

```
    "certificatePem": "
-----BEGIN CERTIFICATE-----
MIICiTCCEXAMPLE6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgEXAMPLEAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSEXAMPLE2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYEXAMPLEb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCEXAMPLEJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDAEXAMPLEsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEXAMPLE25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+aEXAMPLE
EXAMPLEfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZEXAMPLELG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAEXAMPLEWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9qEXAMPLEyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDEXAMPLEBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----\n",
    "keyPair": {
        "PublicKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkEXAMPLEQEFAAOCAQ8AMIIBCgKCAQEAEXAMPLE1nnyJwKSMHw4h\nMMEXAMPLEuuN/
dMAS3fyce8DW/4+EXAMPLEyjmoF/YVF/gHr99VEEXAMPLE5VF13\n59VK7cEXAMPLE67GK+y+jikqXOgHh/
xJTwo
+sGpWEXAMPLEDz18xOd2ka4tCzuWEXAMPLEahJbYkCPUBSU8opVkR7qkEXAMPLE1DR6sx2HocliOOLtu6Fkw91swQWEX
\GB3ZPrNh0PzQYvjUStZeccyNCx2EXAMPLEvp9mQOUXP6plfgxwKRX2fEXAMPLEDa
\nhJLXkX3rHU2xbxJSq7D+XEXAMPLEcw+LyFhI5mgFRl88eGdsAEXAMPLElnI9EesG\nFQIDAQAB\n-----
END PUBLIC KEY-----\n",
        "PrivateKey": "-----BEGIN RSA PRIVATE KEY-----\nkey omittted for security
 reasons\n-----END RSA PRIVATE KEY-----\n"
    }
}
```

자세한 내용은 AWS IoT 개발자 안내서의 IoT 디바이스 인증서 생성 및 등록을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조CreateKeysAndCertificate의 섹션을 참조하세요. AWS CLI

### `create-mitigation-action`

다음 코드 예시에서는 `create-mitigation-action`을 사용하는 방법을 보여 줍니다.

AWS CLI

완화 조치를 생성하려면

다음 create-mitigation-action 예제에서는 적용 AddThingsToQuarantineGroup1Action 시 사물을 라는 사물 그룹으로 이동하는 라는 완화 작업을 정의합니다QuarantineGroup1. 이 작업은 동적 사물 그룹을 재정의합니다.

```
aws iot create-mitigation-action --cli-input-json file::params.json
```

params.json의 콘텐츠:

```
{
    "actionName": "AddThingsToQuarantineGroup1Action",
    "actionParams": {
        "addThingsToThingGroupParams": {
            "thingGroupNames": [
                "QuarantineGroup1"
            ],
            "overrideDynamicGroups": true
        }
    },
    "roleArn": "arn:aws:iam::123456789012:role/service-role/
MoveThingsToQuarantineGroupRole"
}
```

출력:

```
{
    "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
AddThingsToQuarantineGroup1Action",
    "actionId": "992e9a63-a899-439a-aa50-4e20c52367e1"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 CreateMitigationAction (완화 작업 명령)을 참조하세요.

- 자세한 API 내용은 명령 참조CreateMitigationAction의 섹션을 참조하세요. AWS CLI

### create-ota-update

다음 코드 예시에서는 create-ota-update을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon Free와 함께 사용할 OTA 업데이트를 생성하려면RTOS

다음 `create-ota-update` 예제에서는 사물 또는 그룹의 대상 그룹에 OTAUpdate AWS IoT를 생성합니다. 이는 단일 디바이스 또는 디바이스 그룹에 새 펌웨어 이미지를 배포할 수 있는 Amazon FreeRTOS over-the-air 업데이트의 일부입니다.

```
aws iot create-ota-update \
    --cli-input-json file://create-ota-update.json
```

`create-ota-update.json`의 콘텐츠:

```
{
    "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"
}
```

출력:

```
{
```

```
    "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"
}
```

자세한 내용은 AWS IoT API 참조의 CreateOTAUpdate를 참조하세요.

- 자세한 API 내용은 명령 참조CreateOtaUpdate의 섹션을 참조하세요. AWS CLI

### create-policy-version

다음 코드 예시에서는 create-policy-version을 사용하는 방법을 보여 줍니다.

AWS CLI

새 버전으로 정책을 업데이트하려면

다음 create-policy-version 예제에서는 정책 정의를 업데이트하여 새 정책 버전을 생성합니다. 또한 이 예제에서는 새 버전을 기본값으로 설정합니다.

```
aws iot create-policy-version \
    --policy-name UpdateDeviceCertPolicy \
    --policy-document file://policy.json \
    --set-as-default
```

policy.json의 콘텐츠:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action":  "iot:UpdateCertificate",
            "Resource": "*"
        }
    ]
}
```

출력:

```
{
    "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
}
```

자세한 내용은 AWS IoT 개발자 안내서의 IoT 정책을 AWS 참조하세요 IoT.

- 자세한 API 내용은 명령 참조CreatePolicyVersion의 섹션을 참조하세요. AWS CLI

### `create-policy`

다음 코드 예시에서는 `create-policy`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS IoT 정책을 생성하려면

다음 `create-policy` 예제에서는 이라는 AWS IoT 정책을 생성합니다 TemperatureSensorPolicy. `policy.json` 파일에는 AWS IoT 정책 작업을 허용하는 문이 포함되어 있습니다.

```
aws iot create-policy \
    --policy-name TemperatureSensorPolicy \
    --policy-document file://policy.json
```

`policy.json`의 콘텐츠:

```
{
    "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"
            ]
        }
    ]
}
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 IoT 정책을 AWS 참조하세요 IoT.

- 자세한 API 내용은 명령 참조CreatePolicy의 섹션을 참조하세요. AWS CLI

### `create-provisioning-claim`

다음 코드 예시에서는 `create-provisioning-claim`을 사용하는 방법을 보여 줍니다.

AWS CLI

프로비저닝 클레임을 생성하려면

다음 `create-provisioning-claim` 예제에서는 프로비저닝 템플릿에서 프로비저닝 클레임을 생성합니다.

```
aws iot create-provisioning-claim \
    --template-name MyTestProvisioningTemplate
```

출력:


```
{
    "certificateId":
 "78de02184b2ce80cf8fb709bda59e62b19fb83513590483eb0434589476ab09f",
    "certificatePem": "-----BEGIN CERTIFICATE-----\nMIIDdzCCAl
+gAwIBAgIUXSZhEBLztMLZ2fHG
14gV0NymYY0wDQYJKoZIhvcNAQEL
\nBQAwfjELMAkGA1UEBhMCVVMxEzARBgNVBAgMCldhc2hpbmd0b24xEDAOBg
VBAcM\nB1NlYXR0bGUxGDAWBgNVBAoMD0FtYXpvbi5jb20gSW5jLjEgMB4GA1UECwwXQW1h
\nem9uIElvVCBQcm9
2aXNpb25pbmcxDDAKBgNVBAUTAzEuMDAeFw0yMDA3MjgxNjQ0\nMDZaFw0yMDA3MjgxNjUxMDZaMEsxSTBHBgNVB
AMMQDFhNDEyM2VkNmIxYjU3MzE3\nZTgzMTJmY2MzN2FiNTdhY2MzYTZkZGVjOGQ5OGY3NzUwMWRlMjc0YjhmYTQ
xN2Iw\nggEiMA0GCSqGSIb3EXAMPLEAA4IBDwAwggEKAoIBAQDBhKI94ktKLqTwnj+ayOq1\nTAJt/
N6s6IJDZvl
rYjkC0E7wzaeY3TprWk03S29vUzVuEOXHXQXZbihgpg2m6fza\nkWm9/
wpjzE9ny5+xkPGVH4Wnwz7yK5m8S0agL
T96cRBSWnWmonOWdY0GKVzni0CA\n+iyGudgrFKm7Eae/
v18oXrf82KtOAGO4xG0KE2WKYHsT1fx3c9xZhlXP/eX
Lhv00\n+lGp0WVw9PbhKfrxliKJ5q6sL5nVUaUHq6hlQPYwsATeOvAp3u0ak5zgTyL0fg7Y
\nPyKk6VYwLW62r+V
YBSForEMOAhkq3LsP/rjxpEKmi2W4lPVS6oFZRKcD+H1Kyil5\nAgMBAAGjIDAeMAwGA1UdEwEB/
wQCMAAwDgYDV
R0PAQH/BAQDAgeAMA0GCSqGSIb3\nDQEBCwUAA4IBAQAGgix2k6nVqbZFKq97/fZBzLGS0dyz5rT/
E41cDIRX+1j
EPW41\nw0D+2sXheCZLZZnSkvIiP74IToNeXDrjdcaodeGFVHIElRjhMIq+4ZebPbRLtidF
\nRc2hfcTAlqq9Z6v
5Vk6BeM1tu0RqH1wPoVUccLPya8EjNCbnJZUmGdOfrN/Y9pho\n5ikV+HPeZhG/k6dhE2GsQJyKFVHL/
uBgKSily
1bRyWU1r6qcpWBNBHjUoD7HgOwD
\nnzMh4XRb2FQDsqFalkCSYmeL8IVC49sgPD9Otyp5uteGMTy62usAAUQdq/f
ZvrWg\nOkFpwMVnGKVKT7Kg0kKOLzKWOBB2Jm4/gmrJ\n-----END CERTIFICATE-----\n",
    "keyPair": {
        "PublicKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCg
KCAQEAwYSiPeJLSi6k8J4/msjq
\ntUwCbfzerOiCQ2b5a2I5AtBO8M2nmN06a1pNN0tvb1M1bhDlx10F2W4oYKYN
pun8\n2pFpvf8KY8xPZ8ufsZDxlR+Fp8M+8iuZvEtGoC0/enEQUlp1pqJzlnWNBilc54tA
\ngPoshrnYKxSpuxGn
v79fKF63/NirTgBjuMRtChNlimEXAMPLE3PcWYZVz/3ly4b9\nNPpRqdFlcPT24Sn68ZYiieaurC
+Z1VGlB6uoZU
D2MLAE3jrwKd7tGpOc4E8i9H4O\n2D8ipOlWMC1utq/
lWAUhaKxDDgIZKty7D/648aRCpotluJT1UuqBWUSnA/h9
Ssop\neQIDAQAB\n-----END PUBLIC KEY-----\n",
```

```
        "PrivateKey": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEowIBAAKCAQEAwYSiPeJLSi6k8J4/
msjqtUwCbfzerOiCQ2b5a2I5AtBO8M2n
\nmN06a1pNN0tvb1M1bhDlx10F2W4oYKYNpun82pFpvf8KY8xPZ8ufsZ
DxlR+Fp8M+\n8iuZvEtGoC0/enEQUlp1pqJzlnWNBilc54tAgPoshrnYKxSpuxGnv79fKF63/Nir
\nTgBjuMRtCh
NlimB7E9X8d3PcWYZVz/3ly4b9NPpRqdFlcPT24Sn68ZYiieaurC+Z
\n1VGlB6uoZUD2MLAE3jrwKd7tGpOc4E8i
9H4O2D8ipOlWMC1utq/lWAUhaKxDDgIZ\nKty7D/648aRCpotluJT1UuqBWUSnA/
h9SsopeQIDAQABAoIBAEAybN
QUtx9T2/nK\ntZT2pA4iugecxI4dz+DmT0XVXs5VJmrx/
nBSq6ejXExEpSIMO4RY7LE3ZdJcnd56\nF7tQkkY7yR
VzfxHeXFU1krOIPuxWebNOrRoPZr+1RSer+wv2aBC525+88pVuR6tM
\nm3pgkrR2ycCj9FdOUoQxdjHBHaM5PDmJ
9aSxCKdg3nReepeGwsR2TQA+m2vVxWk7\nou0+91eTOP+/QfP7P8ZjOIkO2XivlRcVDyN/
E4QXPKuIkM/8vS8VK+
E9pATQ0MtB\n2lw8R/YU5AJd6jlEXAMPLEGU2UzRzInNWiLtkPPPqgqXXhxOf+mxByjcMalVJk0L
\nhOG2ROUCgY
EA+ROcHNHy/XbsP7FihOhEh+6Q2QxQ2ncBUPYbBazrR8Hn+7SCICQK
\nVyYfd8Ajfq3e7RsKVL5SlMBp7Slidxak
bIn28fKfPn62DaemGCIoyDgLpF+eUxBx
\ngzbCiBZga8brfurza43UZjKZLpg3hq721+FeAiXi1Nma4Yr9YWEHEN
8CgYEAxuWt\npzdWWmsiFzfsAw0sy9ySDA/xr5WRWzJyAqUsjsks6rxNzWebpufnYHcmtW7pLdqM
\nkboHwN2pXa
kmZvrk2nKkEMq5brBYGDXuxDe+V369Bianx8aZFyIsckA7OwXW1w1h
\ngRC5rQ4XOgp3+Jmw7eAO8LRYDjaN846+
QbtO2KcCgYAWS0UL51bijQR0ZwI0dz27\nFQVuCAYsp748aurcRTACCj8jbnK/
QbqTNlxWsaH7ssBjZKo2D5sAqY
BRtASWODab\naHXsDhVm2Jye+ESLoHMaCLoyCkT3ll8yqXIcEDStMO7fO1Ryag164EiJvSIrMfny\nNL/
fXVjCSH
/udCxdzPt+7QKBgQC+LAD7rxdr4J9538hTqpc4XK9vxRbrMXEH55XH
\nHbMa2xONZXpmeTgEQBukyohCVceyRhK9
i0e6irZTjVXghOeoTpC8VXkzcnzouTiQ
\neFQQSGfnp7Ioe6UIz23715pKduzSNkMSKrG924ktv7CyDBF1gBQI5g
aDoHnddJBJ\nPRTIZQKBgA8MASXtTxQntRwXXzR92U0vAighiuRkB/mx9jQpUcK1qiqHbkAMqgNF
\nPFCBYIUbFT
iYKKKeJNbyJQvjfsJCkAnaFJ+RnTxk0Q6Wjm20peJ/ii4QiDdnigoE\nvdlc5cFQewWb4/
zqAtPdinkPlN94ileI
79XQdc7RlJ0jpgTimL+V\n-----END RSA PRIVATE KEY-----\n"
    },
    "expiration": 1595955066.0
}
```

자세한 내용은 IoT Core 개발자 안내서의 신뢰할 수 있는 사용자별 프로비저닝을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조CreateProvisioningClaim의 섹션을 참조하세요. AWS CLI

### `create-provisioning-template-version`

다음 코드 예시에서는 `create-provisioning-template-version`을 사용하는 방법을 보여 줍니다.

AWS CLI

프로비저닝 템플릿 버전을 생성하려면

다음 예제에서는 지정된 프로비저닝 템플릿에 대한 버전을 생성합니다. 새 버전의 본문은 파일 에 제공됩니다`template.json`.

```
aws iot create-provisioning-template-version \
    --template-name widget-template \
    --template-body file://template.json
```

`template.json`의 콘텐츠:

```
{
    "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"]}
        }
    }
```

```
}
```

출력:

```
{
    "templateArn": "arn:aws:iot:us-east-1:123456789012:provisioningtemplate/widget-
template",
    "templateName": "widget-template",
    "versionId": 2,
    "isDefaultVersion": false
}
```

자세한 내용은 AWS IoT Core 개발자 안내서의 IoT Secure Tunneling을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조CreateProvisioningTemplateVersion의 섹션을 참조하세요. AWS CLI

### `create-provisioning-template`

다음 코드 예시에서는 `create-provisioning-template`을 사용하는 방법을 보여 줍니다.

AWS CLI

프로비저닝 템플릿을 생성하려면

다음 `create-provisioning-template` 예제에서는 파일 에 정의된 프로비저닝 템플릿을 생성합니다`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
```

`template.json`의 콘텐츠:

```
{
    "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"]}
        }
    }
}
```

출력:

```
{
    "templateArn": "arn:aws:iot:us-east-1:123456789012:provisioningtemplate/widget-
template",
    "templateName": "widget-template",
    "defaultVersionId": 1
}
```

자세한 내용은 AWS IoT Core 개발자 안내서의 IoT Secure Tunneling을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조CreateProvisioningTemplate의 섹션을 참조하세요. AWS CLI

### create-role-alias

다음 코드 예시에서는 create-role-alias을 사용하는 방법을 보여 줍니다.

AWS CLI

역할 별칭을 생성하려면

다음 create-role-alias 예제에서는 지정된 역할에 LightBulbRole 대해 라는 역할 별칭을 생성합니다.

```
aws iot create-role-alias \
    --role-alias LightBulbRole \
```

```
    --role-arn arn:aws:iam::123456789012:role/lightbulbrole-001
```

출력:

```
{
    "roleAlias": "LightBulbRole",
    "roleAliasArn": "arn:aws:iot:us-west-2:123456789012:rolealias/LightBulbRole"
}
```

자세한 내용은 IoT 참조CreateRoleAlias의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조CreateRoleAlias의 섹션을 참조하세요. AWS CLI

### `create-scheduled-audit`

다음 코드 예시에서는 `create-scheduled-audit`을 사용하는 방법을 보여 줍니다.

AWS CLI

예약된 감사를 생성하려면

다음 `create-scheduled-audit` 예제에서는 매주 수요일에 실행되는 예약된 감사를 생성하여 CA 인증서 또는 디바이스 인증서가 만료되는지 확인합니다.

```
aws iot create-scheduled-audit \
    --scheduled-audit-name WednesdayCertCheck \
    --frequency WEEKLY \
    --day-of-week WED \
    --target-check-
names CA_CERTIFICATE_EXPIRING_CHECK DEVICE_CERTIFICATE_EXPIRING_CHECK
```

출력:

```
{
    "scheduledAuditArn": "arn:aws:iot:us-west-2:123456789012:scheduledaudit/
WednesdayCertCheck"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 감사 명령을 참조하세요.

- 자세한 API 내용은 명령 참조CreateScheduledAudit의 섹션을 참조하세요. AWS CLI

### create-security-profile

다음 코드 예시에서는 create-security-profile을 사용하는 방법을 보여 줍니다.

AWS CLI

보안 프로필을 생성하려면

다음 create-security-profile 예제에서는 셀룰러 대역폭이 임계값을 초과하는지 또는 5분 기간 내에 10회 이상의 권한 부여 실패가 발생하는지 확인하는 보안 프로파일을 생성합니다.

```
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}}]"
```

출력:

```
{
    "securityProfileName": "PossibleIssue",
    "securityProfileArn": "arn:aws:iot:us-west-2:123456789012:securityprofile/
PossibleIssue"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 Detect Commands를 참조하세요.

- 자세한 API 내용은 명령 참조CreateSecurityProfile의 섹션을 참조하세요. AWS CLI

### create-stream

다음 코드 예시에서는 create-stream을 사용하는 방법을 보여 줍니다.

AWS CLI

를 통해 청크로 하나 이상의 대용량 파일을 전송하기 위한 스트림을 생성하려면 MQTT

다음 create-stream 예제에서는 를 통해 청크로 하나 이상의 대용량 파일을 전송하기 위한 스트림을 생성합니다MQTT. 스트림은 S3와 같은 소스의 MQTT 메시지로 패키징된 청크 또는 블록으로 데이터 바이트를 전송합니다. 스트림 1개에 다수의 파일을 연결할 수 있습니다.

```
aws iot create-stream \
    --cli-input-json file://create-stream.json
```

create-stream.json의 콘텐츠:

```
{
    "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"
}
```

출력:

```
{
     "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"
}
```

자세한 내용은 IoT 참조CreateStream의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조CreateStream의 섹션을 참조하세요. AWS CLI

### create-thing-group

다음 코드 예시에서는 create-thing-group을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 사물 그룹을 생성하려면

다음 create-thing-group 예제에서는 설명과 두 개의 속성이 LightBulbs 있는 사물 그룹을 생성합니다.

```
aws iot create-thing-group \
    --thing-group-name LightBulbs \
    --thing-group-properties "thingGroupDescription=\"Generic bulb group\",
 attributePayload={attributes={Manufacturer=AnyCompany,wattage=60}}"
```

출력:

```
{
    "thingGroupName": "LightBulbs",
    "thingGroupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs",
    "thingGroupId": "9198bf9f-1e76-4a88-8e8c-e7140142c331"
}
```

예제 2: 상위 그룹의 일부인 사물 그룹을 생성하려면

다음은 라는 상위 사물 그룹이 HalogenBulbs 있는 라는 사물 그룹을 create-thing-group 생성합니다LightBulbs.

```
aws iot create-thing-group \
    --thing-group-name HalogenBulbs \
    --parent-group-name LightBulbs
```

출력:

```
{
    "thingGroupName": "HalogenBulbs",
    "thingGroupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/HalogenBulbs",
    "thingGroupId": "f4ec6b84-b42b-499d-9ce1-4dbd4d4f6f6e"
}
```

자세한 내용은 IoT 개발자 안내서의 사물 그룹을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조CreateThingGroup의 섹션을 참조하세요. AWS CLI

### create-thing-type

다음 코드 예시에서는 create-thing-type을 사용하는 방법을 보여 줍니다.

AWS CLI

사물 유형을 정의하려면

다음 create-thing-type 예제에서는 사물 유형과 관련 속성을 정의합니다.

```
aws iot create-thing-type \
    --thing-type-name "LightBulb" \
    --thing-type-properties "thingTypeDescription=light bulb type,
 searchableAttributes=wattage,model"
```

출력:

```
{
    "thingTypeName": "LightBulb",
    "thingTypeArn": "arn:aws:iot:us-west-2:123456789012:thingtype/LightBulb",
    "thingTypeId": "ce3573b0-0a3c-45a7-ac93-4e0ce14cd190"
}
```

자세한 내용은 IoT 개발자 안내서의 사물 유형을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조CreateThingType의 섹션을 참조하세요. AWS CLI

### create-thing

다음 코드 예시에서는 create-thing을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 레지스트리에서 사물 레코드 생성

다음 create-thing 예제에서는 AWS IoT 사물 레지스트리에서 디바이스에 대한 항목을 생성합니다.

```
aws iot create-thing \
    --thing-name SampleIoTThing
```

출력:

```
{
    "thingName": "SampleIoTThing",
    "thingArn": "arn:aws:iot:us-west-2: 123456789012:thing/SampleIoTThing",
    "thingId": " EXAMPLE1-90ab-cdef-fedc-ba987EXAMPLE "
}
```

예제 2: 사물 유형과 연결된 사물을 정의하려면

다음 `create-thing` 예제에서는 지정된 사물 유형과 속성이 있는 사물을 생성합니다.

```
aws iot create-thing \
    --thing-name "MyLightBulb" \
    --thing-type-name "LightBulb" \
    --attribute-payload "{"attributes": {"wattage":"75", "model":"123"}}"
```

출력:

```
{
    "thingName": "MyLightBulb",
    "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyLightBulb",
    "thingId": "40da2e73-c6af-406e-b415-15acae538797"
}
```

자세한 내용은 IoT 개발자 안내서의 레지스트리 및 사물 유형을 사용하여 사물을 관리하는 방법을 참조하세요. https://docs.aws.amazon.com/iot/latest/developerguide/thing-types.html AWS IoT

- 자세한 API 내용은 명령 참조CreateThing의 섹션을 참조하세요. AWS CLI

### `create-topic-rule-destination`

다음 코드 예시에서는 `create-topic-rule-destination`을 사용하는 방법을 보여 줍니다.

AWS CLI

주제 규칙 대상을 생성하려면

다음 `create-topic-rule-destination` 예제에서는 HTTP 엔드포인트에 대한 주제 규칙 대상을 생성합니다.

```
aws iot create-topic-rule-destination \
    --destination-configuration httpUrlConfiguration={confirmationUrl=https://
example.com}
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 AWS IoT 개발자 안내서의 주제 규칙 대상 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateTopicRuleDestination의 섹션을 참조하세요. AWS CLI

### create-topic-rule

다음 코드 예시에서는 create-topic-rule을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon SNS 알림을 보내는 규칙을 생성하려면

다음 create-topic-rule 예제에서는 디바이스 섀도우에서 볼 수 있듯이 오염 수분 수준 판독값이 낮을 때 Amazon SNS 메시지를 보내는 규칙을 생성합니다.

```
aws iot create-topic-rule \
    --rule-name "LowMoistureRule" \
    --topic-rule-payload file://plant-rule.json
```

이 예제에서는 다음 JSON 코드를 라는 파일에 저장해야 합니다plant-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"
            }
    }]
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 IoT 규칙 생성을 AWS 참조하세요 IoT.

- 자세한 API 내용은 명령 참조CreateTopicRule의 섹션을 참조하세요. AWS CLI

### `delete-account-audit-configuration`

다음 코드 예시에서는 `delete-account-audit-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 계정에 대한 모든 감사 확인을 비활성화하려면

다음 `delete-account-audit-configuration` 예제에서는 이 계정의 AWS IoT Device Defender 기본 설정을 복원하여 모든 감사 검사를 비활성화하고 구성 데이터를 삭제합니다. 또한 이 계정에 대해 예약된 감사도 삭제합니다. 이 명령은 주의하여 사용합니다.

```
aws iot delete-account-audit-configuration \
    --delete-scheduled-audits
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 감사 명령을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteAccountAuditConfiguration의 섹션을 참조하세요. AWS CLI

### delete-audit-suppression

다음 코드 예시에서는 delete-audit-suppression을 사용하는 방법을 보여 줍니다.

AWS CLI

감사 결과 금지를 삭제하려면

다음 delete-audit-suppression 예제에서는 DEVICE_CERTIFICATE_EXPIRING_에 대한 감사 결과 금지를 삭제합니다CHECK.

```
aws iot delete-audit-suppression \
    --check-name DEVICE_CERTIFICATE_EXPIRING_CHECK \
    --resource-identifier deviceCertificateId="c7691e<shortened>"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IoT 개발자 안내서의 감사 결과 금지를 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조DeleteAuditSuppression의 섹션을 참조하세요. AWS CLI

### delete-authorizer

다음 코드 예시에서는 delete-authorizer을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 권한 부여자를 삭제하려면

다음 delete-authorizer 예제에서는 이름이 인 권한 부여자를 삭제합니다CustomAuthorizer. 사용자 지정 권한 부여자는 삭제하기 전에 INACTIVE 상태에 있어야 합니다.

```
aws iot delete-authorizer \
    --authorizer-name CustomAuthorizer
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IoT 개발자 안내서DeleteAuthorizer의 섹션을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조DeleteAuthorizer의 섹션을 참조하세요. AWS CLI

### delete-billing-group

다음 코드 예시에서는 delete-billing-group을 사용하는 방법을 보여 줍니다.

AWS CLI

결제 그룹을 삭제하려면

다음 delete-billing-group 예제에서는 지정된 결제 그룹을 삭제합니다. 결제 그룹에 하나 이상의 사물이 포함되어 있더라도 결제 그룹을 삭제할 수 있습니다.

```
aws iot delete-billing-group \
    --billing-group-name BillingGroupTwo
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IoT 개발자 안내서의 결제 그룹을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조DeleteBillingGroup의 섹션을 참조하세요. AWS CLI

### delete-ca-certificate

다음 코드 예시에서는 delete-ca-certificate을 사용하는 방법을 보여 줍니다.

AWS CLI

CA 인증서를 삭제하려면

다음 delete-ca-certificate 예제에서는 지정된 인증서 ID가 있는 CA 인증서를 삭제합니다.

```
aws iot delete-ca-certificate \
    --certificate-
id f4efed62c0142f16af278166f61962501165c4f05362952074264600 58cd1467
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT API 참조의 DeleteCACertificate를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteCaCertificate의 섹션을 참조하세요. AWS CLI

### delete-certificate

다음 코드 예시에서는 delete-certificate을 사용하는 방법을 보여 줍니다.

AWS CLI

디바이스 인증서를 삭제하려면

다음 delete-certificate 예제에서는 지정된 ID가 있는 디바이스 인증서를 삭제합니다.

```
aws iot delete-certificate \
    --certificate-
id c0c57bbc8baaf4631a9a0345c957657f5e710473e3ddbee1428d216d54d53ac9
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IoT 참조DeleteCertificate의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조DeleteCertificate의 섹션을 참조하세요. AWS CLI

### delete-custom-metric

다음 코드 예시에서는 delete-custom-metric을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 지표를 삭제하려면

다음 delete-custom-metric 예제에서는 사용자 지정 지표를 삭제합니다.

```
aws iot delete-custom-metric \
    --metric-name batteryPercentage \
    --region us-east-1
```

출력:

```
HTTP 200
```

자세한 내용은 AWS IoT Core 개발자 안내서의 사용자 지정 지표를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteCustomMetric의 섹션을 참조하세요. AWS CLI

### delete-dimension

다음 코드 예시에서는 delete-dimension을 사용하는 방법을 보여 줍니다.

AWS CLI

차원을 삭제하려면

다음 delete-dimension 예제에서는 이라는 차원을 삭제합니
다TopicFilterForAuthMessages.

```
aws iot delete-dimension \
    --name TopicFilterForAuthMessages
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 Detect Commands를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteDimension의 섹션을 참조하세요. AWS CLI

### delete-domain-configuration

다음 코드 예시에서는 delete-domain-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

도메인 구성을 삭제하려면

다음 delete-domain-configuration 예제에서는 AWS 계정additionalDataDomain에서
라는 도메인 구성을 삭제합니다.

```
aws iot delete-domain-configuration \
    --domain-configuration-name "additionalDataDomain" \
    --domain-configuration-status "OK"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 구성 가능한 엔드포인트를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteDomainConfiguration의 섹션을 참조하세요. AWS CLI

### delete-dynamic-thing-group

다음 코드 예시에서는 delete-dynamic-thing-group을 사용하는 방법을 보여 줍니다.

AWS CLI

동적 사물 그룹을 삭제하려면

다음 delete-dynamic-thing-group 예제에서는 지정된 동적 사물 그룹을 삭제합니다.

```
aws iot delete-dynamic-thing-group \
    --thing-group-name "RoomTooWarm"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IoT 개발자 안내서의 동적 사물 그룹을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조DeleteDynamicThingGroup의 섹션을 참조하세요. AWS CLI

### delete-job-execution

다음 코드 예시에서는 delete-job-execution을 사용하는 방법을 보여 줍니다.

AWS CLI

작업 실행을 삭제하려면

다음 delete-job-execution 예제에서는 디바이스에서 지정된 작업의 작업 실행을 삭제합니다. describe-job-execution 를 사용하여 실행 번호를 가져옵니다.

```
aws iot delete-job-execution
    --job-id "example-job-02"
    --thing-name "MyRaspberryPi"
    --execution-number 1
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 작업 생성 및 관리(CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteJobExecution의 섹션을 참조하세요. AWS CLI

### delete-job

다음 코드 예시에서는 delete-job을 사용하는 방법을 보여 줍니다.

AWS CLI

작업을 삭제하려면

다음 delete-job 예제에서는 지정된 작업을 삭제합니다. --force 옵션을 지정하면 상태가 인 경우에도 작업이 삭제됩니다IN_PROGRESS.

```
aws iot delete-job \
    --job-id "example-job-04" \
    --force
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 작업 생성 및 관리(CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteJob의 섹션을 참조하세요. AWS CLI

### delete-mitigation-action

다음 코드 예시에서는 delete-mitigation-action을 사용하는 방법을 보여 줍니다.

AWS CLI

완화 작업을 삭제하려면

다음 delete-mitigation-action 예제에서는 지정된 완화 작업을 삭제합니다.

```
aws iot delete-mitigation-action \
    --action-name AddThingsToQuarantineGroup1Action
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 DeleteMitigationAction (완화 작업 명령)을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteMitigationAction의 섹션을 참조하세요. AWS CLI

### delete-ota-update

다음 코드 예시에서는 delete-ota-update을 사용하는 방법을 보여 줍니다.

AWS CLI

OTA 업데이트를 삭제하려면

다음 delete-ota-update 예제에서는 지정된 OTA 업데이트를 삭제합니다.

```
aws iot delete-ota-update \
    --ota-update-id ota12345 \
    --delete-stream \
    --force-delete-aws-job
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT API 참조의 DeleteOTAUpdate를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteOtaUpdate의 섹션을 참조하세요. AWS CLI

### delete-policy-version

다음 코드 예시에서는 delete-policy-version을 사용하는 방법을 보여 줍니다.

AWS CLI

정책 버전을 삭제하려면

다음 delete-policy-version 예제에서는 AWS 계정에서 지정된 정책의 버전 2를 삭제합니다.

```
aws iot delete-policy-version \
    --policy-name UpdateDeviceCertPolicy \
    --policy-version-id 2
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 IoT 정책을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조DeletePolicyVersion의 섹션을 참조하세요. AWS CLI

### delete-policy

다음 코드 예시에서는 delete-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

정책을 삭제하는 방법

다음 delete-policy 예제에서는 AWS 계정에서 지정된 정책을 삭제합니다.

```
aws iot delete-policy --policy-name UpdateDeviceCertPolicy
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 IoT 정책을 AWS 참조하세요 IoT.

- 자세한 API 내용은 명령 참조DeletePolicy의 섹션을 참조하세요. AWS CLI

### delete-provisioning-template-version

다음 코드 예시에서는 delete-provisioning-template-version을 사용하는 방법을 보여 줍니다.

AWS CLI

프로비저닝 템플릿 버전을 삭제하려면

다음 delete-provisioning-template-version 예제에서는 지정된 프로비저닝 템플릿의 버전 2를 삭제합니다.

```
aws iot delete-provisioning-template-version \
    --version-id 2 \
    --template-name "widget-template"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT Core 개발자 안내서의 IoT Secure Tunneling을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조DeleteProvisioningTemplateVersion의 섹션을 참조하세요. AWS CLI

### delete-provisioning-template

다음 코드 예시에서는 delete-provisioning-template을 사용하는 방법을 보여 줍니다.

AWS CLI

프로비저닝 템플릿을 삭제하려면

다음 delete-provisioning-template 예제에서는 지정된 프로비저닝 템플릿을 삭제합니다.

```
aws iot delete-provisioning-template \
```

```
    --template-name widget-template
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT Core 개발자 안내서의 IoT 보안 터널링을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조DeleteProvisioningTemplate의 섹션을 참조하세요. AWS CLI

### delete-registration-code

다음 코드 예시에서는 delete-registration-code을 사용하는 방법을 보여 줍니다.

AWS CLI

등록 코드 삭제

다음 delete-registration-code 예제에서는 AWS IoT 계정별 등록 코드를 삭제합니다.

```
aws iot delete-registration-code
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 자체 인증서 사용을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteRegistrationCode의 섹션을 참조하세요. AWS CLI

### delete-role-alias

다음 코드 예시에서는 delete-role-alias을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS IoT 역할 별칭을 삭제하려면

다음 delete-role-alias 예제에서는 이름이 인 AWS IoT 역할 별칭을 삭제합니다LightBulbRole.

```
aws iot delete-role-alias \
    --role-alias LightBulbRole
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 AWS 서비스에 대한 직접 호출 승인을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteRoleAlias의 섹션을 참조하세요. AWS CLI

### delete-scheduled-audit

다음 코드 예시에서는 delete-scheduled-audit을 사용하는 방법을 보여 줍니다.

AWS CLI

예약된 감사를 삭제하려면

다음 delete-scheduled-audit 예제에서는 라는 AWS IoT Device Defender 예약 감사를 삭제합니다AWSIoTDeviceDefenderDailyAudit.

```
aws iot delete-scheduled-audit \
    --scheduled-audit-name AWSIoTDeviceDefenderDailyAudit
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 감사 명령을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteScheduledAudit의 섹션을 참조하세요. AWS CLI

### delete-security-profile

다음 코드 예시에서는 delete-security-profile을 사용하는 방법을 보여 줍니다.

AWS CLI

보안 프로필을 삭제하려면

다음 delete-security-profile 예제에서는 라는 보안 프로파일을 삭제합니다PossibleIssue.

```
aws iot delete-security-profile \
    --security-profile-name PossibleIssue
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 Detect Commands를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteSecurityProfile의 섹션을 참조하세요. AWS CLI

### delete-stream

다음 코드 예시에서는 delete-stream을 사용하는 방법을 보여 줍니다.

AWS CLI

스트림을 삭제하려면

다음 delete-stream 예제에서는 지정된 스트림을 삭제합니다.

```
aws iot delete-stream \
    --stream-id stream12345
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IoT 참조DeleteStream의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조DeleteStream의 섹션을 참조하세요. AWS CLI

### delete-thing-group

다음 코드 예시에서는 delete-thing-group을 사용하는 방법을 보여 줍니다.

AWS CLI

사물 그룹을 삭제하려면

다음 delete-thing-group 예제에서는 지정된 사물 그룹을 삭제합니다. 하위 사물 그룹이 포함된 사물 그룹은 삭제할 수 없습니다.

```
aws iot delete-thing-group \
    --thing-group-name DefectiveBulbs
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IoT 개발자 안내서의 사물 그룹을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조DeleteThingGroup의 섹션을 참조하세요. AWS CLI

### delete-thing-type

다음 코드 예시에서는 delete-thing-type을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 사물 유형을 삭제하려면

다음 delete-thing-type 예제에서는 더 이상 사용되지 않는 사물 유형을 삭제합니다.

```
aws iot delete-thing-type \
    --thing-type-name "obsoleteThingType"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IoT 개발자 안내서의 사물 유형을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조DeleteThingType의 섹션을 참조하세요. AWS CLI

### delete-thing

다음 코드 예시에서는 delete-thing을 사용하는 방법을 보여 줍니다.

AWS CLI

사물에 대한 세부 정보를 표시하려면

다음 delete-thing 예제에서는 AWS 계정의 AWS IoT 레지스트리에서 사물을 삭제합니다.

aws iot delete-thing --thing-name “FourthBulb

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 레지스트리를 사용하여 사물을 관리하는 방법을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteThing의 섹션을 참조하세요. AWS CLI

### delete-topic-rule-destination

다음 코드 예시에서는 delete-topic-rule-destination을 사용하는 방법을 보여 줍니다.

AWS CLI

주제 규칙 대상을 삭제하려면

다음 delete-topic-rule-destination 예제에서는 지정된 주제 규칙 대상을 삭제합니다.

```
aws iot delete-topic-rule-destination \
    --arn "arn:aws:iot:us-west-2:123456789012:ruledestination/http/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 주제 규칙 대상 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteTopicRuleDestination의 섹션을 참조하세요. AWS CLI

### delete-topic-rule

다음 코드 예시에서는 delete-topic-rule을 사용하는 방법을 보여 줍니다.

AWS CLI

규칙을 삭제하려면

다음 delete-topic-rule 예제에서는 지정된 규칙을 삭제합니다.

```
aws iot delete-topic-rule \
    --rule-name "LowMoistureRule"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IoT 개발자 안내서의 규칙 삭제를 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조DeleteTopicRule의 섹션을 참조하세요. AWS CLI

### delete-v2-logging-level

다음 코드 예시에서는 delete-v2-logging-level을 사용하는 방법을 보여 줍니다.

AWS CLI

사물 그룹의 로깅 수준을 삭제하려면

다음 delete-v2-logging-level 예제에서는 지정된 사물 그룹에 대한 로깅 수준을 삭제합니다.

```
aws iot delete-v2-logging-level \
    --target-type THING_GROUP \
```

```
    --target-name LightBulbs
```

이 명령은 출력을 생성하지 않습니다.

- API 자세한 내용은 AWS CLI 명령 참조의 DeleteV2LoggingLevel를 참조하세요.

### deprecate-thing-type

다음 코드 예시에서는 deprecate-thing-type을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 사물 유형 사용 중단

다음 deprecate-thing-type 예제에서는 사용자가 새 사물을 연결할 수 없도록 사물 유형을 사용 중지합니다.

```
aws iot deprecate-thing-type \
    --thing-type-name "obsoleteThingType"
```

이 명령은 출력을 생성하지 않습니다.

예제 2: 사물 유형의 사용 중지를 되돌리려면

다음 deprecate-thing-type 예제에서는 사물 유형의 사용 중지를 되돌리므로 사용자가 새 사물을 다시 연결할 수 있습니다.

```
aws iot deprecate-thing-type \
    --thing-type-name "obsoleteThingType" \
    --undo-deprecate
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IoT 개발자 안내서의 사물 유형을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조DeprecateThingType의 섹션을 참조하세요. AWS CLI

### describe-account-audit-configuration

다음 코드 예시에서는 describe-account-audit-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 감사 구성 설정을 보려면

다음 `describe-account-audit-configuration` 예제에서는 AWS IoT Device Defender 감사 구성의 현재 설정을 나열합니다.

```
aws iot describe-account-audit-configuration
```

출력:

```
{
    "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
        }
    }
}
```

자세한 내용은 AWS IoT 개발자 안내서의 감사 명령을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeAccountAuditConfiguration의 섹션을 참조하세요. AWS CLI

### `describe-audit-finding`

다음 코드 예시에서는 `describe-audit-finding`을 사용하는 방법을 보여 줍니다.

AWS CLI

감사 결과에 대한 세부 정보를 나열하려면

다음 `describe-audit-finding` 예제에서는 지정된 AWS IoT Device Defender 감사 결과에 대한 세부 정보를 나열합니다. 감사는 여러 결과를 생성할 수 있습니다. `list-audit-findings` 명령을 사용하여 감사 결과 목록을 가져와 를 가져옵니다`findingId`.

```
aws iot describe-audit-finding \
    --finding-id "ef4826b8-e55a-44b9-b460-5c485355371b"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS IoT 개발자 안내서의 감사 결과 확인(감사 명령)을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeAuditFinding의 섹션을 참조하세요. AWS CLI

### describe-audit-mitigation-actions-task

다음 코드 예시에서는 describe-audit-mitigation-actions-task을 사용하는 방법을 보여 줍니다.

AWS CLI

감사 완화 조치 작업의 세부 정보를 표시하려면

다음 describe-audit-mitigation-actions-task 예제에서는 가 결과에 ResetPolicyVersionAction 적용된 지정된 작업에 대한 세부 정보를 보여줍니다. 결과에는 작업이 시작 및 종료된 시간, 대상 조사 결과(및 결과) 및 이 작업의 일부로 적용되는 작업의 정의가 포함됩니다.

```
aws iot describe-audit-mitigation-actions-task \
    --task-id ResetPolicyTask01
```

출력:

```
{
    "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"
                }
            }
        }
    ]
}
```

자세한 내용은 AWS IoT 개발자 안내서의 DescribeAuditMitigationActionsTask (완화 작업 명령)을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeAuditMitigationActionsTask의 섹션을 참조하세요. AWS CLI

### describe-audit-suppression

다음 코드 예시에서는 describe-audit-suppression을 사용하는 방법을 보여 줍니다.

AWS CLI

감사 결과 억제에 대한 세부 정보를 가져오려면

다음 describe-audit-suppression 예제에서는 감사 결과 억제에 대한 세부 정보를 나열합니다.

```
aws iot describe-audit-task \
    --task-id "787ed873b69cb4d6cdbae6ddd06996c5"
```

출력:

```
{
    "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
        }
    }
}
```

자세한 내용은 IoT 개발자 안내서의 감사 결과 금지를 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조DescribeAuditSuppression의 섹션을 참조하세요. AWS CLI

### describe-audit-task

다음 코드 예시에서는 describe-audit-task을 사용하는 방법을 보여 줍니다.

AWS CLI

감사 인스턴스에 대한 정보를 가져오려면

다음 describe-audit-task 예제에서는 AWS IoT Device Defender 감사의 인스턴스에 대한 정보를 가져옵니다. 감사가 완료되면 실행에 대한 요약 통계가 결과에 포함됩니다.

```
aws iot describe-audit-task \
    --task-id a3aea009955e501a31b764abe1bebd3d
```

출력:

```
{
   "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
        }
    }
}
```

자세한 내용은 AWS IoT 개발자 안내서의 감사 명령을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeAuditTask의 섹션을 참조하세요. AWS CLI

### describe-authorizer

다음 코드 예시에서는 describe-authorizer을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 권한 부여자에 대한 정보를 가져오려면

다음 describe-authorizer 예제에서는 지정된 사용자 지정 권한 부여자에 대한 세부 정보를 표시합니다.

```
aws iot describe-authorizer \
    --authorizer-name CustomAuthorizer
```

출력:

```
{
    "authorizerDescription": {
        "authorizerName": "CustomAuthorizer",
        "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/
CustomAuthorizer",
        "authorizerFunctionArn": "arn:aws:lambda:us-
west-2:123456789012:function:CustomAuthorizerFunction",
        "tokenKeyName": "MyAuthToken",
        "tokenSigningPublicKeys": {
            "FIRST_KEY": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA1uJOB4lQPgG/lM6ZfIwo
\nZ+7ENxAio9q6QD4FFqjGZsvjtYwjoe1RKK0U8Eq9xb5O3kRSmyIwTzwzm/f4Gf0Y
\nZUloJ+t3PUUwHrmbYTAgTrCUgRFygjfgVwGCPs5ZAX4Eyqt5cr+AIHIiUDbxSa7p
\nzwOBKPeic0asNJpqT8PkBbRaKyleJh5oo81NDHHmVtbBm5A5YiJjqYXLaVAowKzZ\n
```

```
+GqsNvAQ9Jy1wI2VrEa1OfL8flDB/BJLm7zjpfPOHDJQgID0XnZwAlNnZcOhCwIx\n50g2LW2Oy9R/
dmqtDmJiVP97Z4GykxPvwlYHrUXY0iW1R3AR/Ac1NhCTGZMwVDB1\nlQIDAQAB\n-----END PUBLIC
 KEY-----"
        },
        "status": "ACTIVE",
        "creationDate": 1571245658.069,
        "lastModifiedDate": 1571245658.069
    }
}
```

자세한 내용은 IoT 참조DescribeAuthorizer의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조DescribeAuthorizer의 섹션을 참조하세요. AWS CLI

### `describe-billing-group`

다음 코드 예시에서는 `describe-billing-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

결제 그룹에 대한 정보를 가져오려면

다음 `describe-billing-group` 예제에서는 지정된 결제 그룹에 대한 정보를 가져옵니다.

```
aws iot describe-billing-group --billing-group-name GroupOne
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 IoT 개발자 안내서의 결제 그룹을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조DescribeBillingGroup의 섹션을 참조하세요. AWS CLI

### describe-ca-certificate

다음 코드 예시에서는 describe-ca-certificate을 사용하는 방법을 보여 줍니다.

AWS CLI

CA 인증서에 대한 세부 정보를 가져오려면

다음 describe-ca-certificate 예제에서는 지정된 CA 인증서에 대한 세부 정보를 표시합니다.

```
aws iot describe-ca-certificate \
    --certificate-
id f4efed62c0142f16af278166f61962501165c4f05362952074264600058cd1467
```

출력:

```
{
    "certificateDescription": {
        "certificateArn": "arn:aws:iot:us-west-2:123456789012:cacert/
f4efed62c0142f16af278166f61962501165c4f05362952074264600058cd1467",
        "certificateId":
 "f4efed62c0142f16af278166f61962501165c4f05362952074264600058cd1467",
        "status": "INACTIVE",
        "certificatePem": "-----BEGIN CERTIFICATE-----
\nMIICzzCCAbegEXAMPLEJANVEPWXl8taPMA0GCSqGSIb3DQEBBQUAMB4xCzAJBgNV
\nBAYTAlVTMQ8wDQYDVQQKDAZBbWF6b24wHhcNMTkwOTI0MjEzMTE1WhcNMjkwOTIx
\nMjEzMTE1WjAeMQswCQYDVQQGEwJVUzEPMA0GA1UECgwGQW1hem9uMIIBIjANBgkq
\nhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAzd3R3ioalCS0MhFWfBrVGR036EK07UAf
\nVdz9EXAMPLE1VczICbADnATK522kEIB51/18VzlFtAhQL5V5eybXKnB7QebNer5m
\n4Yibx7shR5oqNzFsrXWxuugN5+w5gEfqNMawOjhF4LsculKG49yuqjcDU19/13ua
\n3B2gxs1Pe7TiWWvUskzxnbO1F2WCshbEJvqY8fIWtGYCjTeJAgQ9hvZx/69XhKen
\nwV9LJwOQxrsUS0Ty8IHwbB8fRy72VM3u7fJoaU+nO4jD5cqaoEPtzoeFUEXAMPLE
\nyVAJpqHwgbYbcUfn7V+AB6yh1+0Fa1rEQGuZDPGyJslxwr5vh8nRewIDAQABoxAw
\nDjAMBgNVHRMEBTADAQH/MA0GCSqGSIb3DQEBBQUAA4IBAQA+3a5CV3IJgOnd0AgI
\nBgVMtmYzTvqAngx26aG9/spvCjXckh2SBF+EcBlCFwH1yakwjJL1dR4yarnrfxgI
\nEqP4AOYVimAVoQ5FBwnloHe16+3qtDiblU9DeXBUCtS55EcfrEXAMPLEYtXdqU5C
\nU9ia4KAjV0dxW1+EFYMwX5eGeb0gDTNHBylV6B/fOSZiQAwDYp4x3B+gAP+a/bWB
\nu1umOqtBdWe6L6/83L+JhaTByqV25iVJ4c/UZUnG8926wUlDM9zQvEXuEVvzZ7+m\n4PSNqst/
nVOvnLpoG4e0WgcJgANuB33CSWtjWSuYsbhmqQRknGhREXAMPLEZT4fm\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
        }
    }
}
```

자세한 내용은 AWS IoT API 참조의 DescribeCACertificate를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeCaCertificate의 섹션을 참조하세요. AWS CLI

### `describe-certificate`

다음 코드 예시에서는 `describe-certificate`을 사용하는 방법을 보여 줍니다.

AWS CLI

인증서에 대한 정보를 가져오려면

다음 `describe-certificate` 예제에서는 지정된 인증서에 대한 세부 정보를 표시합니다.

```
aws iot describe-certificate \
    --certificate-
id "4f0ba725787aa94d67d2fca420eca022242532e8b3c58e7465c7778b443fd65e"
```

출력:

```
{
    "certificateDescription": {
        "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/4f0ba725787aa94d67d2fca420eca022242532e8b3c58e7465c7778b443fd65e",
        "certificateId":
 "4f0ba725787aa94d67d2fca420eca022242532e8b3c58e7465c7778b443fd65e",
        "status": "ACTIVE",
        "certificatePem": "-----BEGIN CERTIFICATE-----
MIICiTEXAMPLEQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBEXAMPLEMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSBDEXAMPLElMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5EXAMPLEcNMTEwNDI1MjA0NTIxWhcN
```

```
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNEXAMPLEdBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBEXAMPLEz
b2xEXAMPLEYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5jb20wgZ8EXAMPLEZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
21uUSfwfEvySWtC2XADZ4nB+BLYEXAMPLEpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7EXAMPLEGBzZswY6786m86gpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFEXAMPLEAtCu4
nUhVVxYUnEXAMPLE8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GEXAMPLEl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----",
        "ownedBy": "123456789012",
        "creationDate": 1541022751.983,
        "lastModifiedDate": 1541022751.983,
        "customerVersion": 1,
        "transferData": {},
        "generationId": "6974fbed-2e61-4114-bc5e-4204cc79b045",
        "validity": {
            "notBefore": 1541022631.0,
            "notAfter": 2524607999.0
        }
    }
}
```

자세한 내용은 IoT 참조DescribeCertificate의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조DescribeCertificate의 섹션을 참조하세요. AWS CLI

### `describe-custom-metric`

다음 코드 예시에서는 `describe-custom-metric`을 사용하는 방법을 보여 줍니다.

AWS CLI

Device Defender 사용자 지정 지표에 대한 정보를 가져오려면

다음 `describe-custom-metric` 예제에서는 라는 사용자 지정 지표에 대한 정보를 가져옵니다myCustomMetric.

```
aws iot describe-custom-metric \
    --metric-name myCustomMetric
```

출력:

```
{
    "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
}
```

자세한 내용은 AWS IoT Core 개발자 안내서의 사용자 지정 지표를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeCustomMetric의 섹션을 참조하세요. AWS CLI

### describe-default-authorizer

다음 코드 예시에서는 describe-default-authorizer을 사용하는 방법을 보여 줍니다.

AWS CLI

기본 사용자 지정 권한 부여자에 대한 정보를 가져오려면

다음 describe-default-authorizer 예제에서는 기본 사용자 지정 권한 부여자에 대한 세부 정보를 표시합니다.

```
aws iot describe-default-authorizer
```

출력:

```
{
    "authorizerName": "CustomAuthorizer",
    "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/
CustomAuthorizer"
}
```

자세한 내용은 IoT 참조DescribeDefaultAuthorizer의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조DescribeDefaultAuthorizer의 섹션을 참조하세요. AWS CLI

### describe-dimension

다음 코드 예시에서는 describe-dimension을 사용하는 방법을 보여 줍니다.

AWS CLI

차원에 대한 정보를 가져오려면

다음 describe-dimension 예제에서는 이라는 차원에 대한 정보를 가져옵니 다TopicFilterForAuthMessages.

```
aws iot describe-dimension \
    --name TopicFilterForAuthMessages
```

출력:

```
{
    "name": "TopicFilterForAuthMessages",
    "arn": "arn:aws:iot:eu-west-2:123456789012:dimension/
TopicFilterForAuthMessages",
    "type": "TOPIC_FILTER",
    "stringValues": [
        "device/+/auth"
    ],
    "creationDate": 1578620223.255,
    "lastModifiedDate": 1578620223.255
}
```

자세한 내용은 AWS IoT 개발자 안내서의 Detect Commands를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeDimension의 섹션을 참조하세요. AWS CLI

### describe-domain-configuration

다음 코드 예시에서는 describe-domain-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

도메인 구성을 설명하려면

다음 describe-domain-configuration 예제에서는 지정된 도메인 구성에 대한 세부 정보를 표시합니다.

```
aws iot describe-domain-configuration \
    --domain-configuration-name "additionalDataDomain"
```

출력:

```
{
    "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
}
```

자세한 내용은 AWS IoT 개발자 안내서의 구성 가능한 엔드포인트를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeDomainConfiguration의 섹션을 참조하세요. AWS CLI

### `describe-endpoint`

다음 코드 예시에서는 `describe-endpoint`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 현재 AWS 엔드포인트 가져오기

다음 `describe-endpoint` 예제에서는 모든 명령이 적용되는 기본 AWS 엔드포인트를 검색합니다.

```
aws iot describe-endpoint
```

출력:

```
{
    "endpointAddress": "abc123defghijk.iot.us-west-2.amazonaws.com"
}
```

자세한 내용은 IoT 개발자 안내서DescribeEndpoint의 섹션을 참조하세요. AWS IoT

예제 2: ATS 엔드포인트 가져오기

다음 describe-endpoint 예제에서는 Amazon Trust Services(ATS) 엔드포인트를 검색합니다.

```
aws iot describe-endpoint \
    --endpoint-type iot:Data-ATS
```

출력:

```
{
    "endpointAddress": "abc123defghijk-ats.iot.us-west-2.amazonaws.com"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 X.509 인증서 및 AWS IoT를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeEndpoint의 섹션을 참조하세요. AWS CLI

### describe-event-configurations

다음 코드 예시에서는 describe-event-configurations을 사용하는 방법을 보여 줍니다.

AWS CLI

게시되는 이벤트 유형을 표시하려면

다음 describe-event-configurations 예제에서는 무언가가 추가, 업데이트 또는 삭제될 때 생성되는 이벤트를 제어하는 구성을 나열합니다.

```
aws iot describe-event-configurations
```

출력:

```
{
    "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
        }
    }
}
```

자세한 내용은 AWS IoT 개발자 안내서의 이벤트 메시지를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeEventConfigurations의 섹션을 참조하세요. AWS CLI

### describe-index

다음 코드 예시에서는 describe-index을 사용하는 방법을 보여 줍니다.

AWS CLI

사물 인덱스의 현재 상태를 검색하려면

다음 describe-index 예제에서는 사물 인덱스의 현재 상태를 검색합니다.

```
aws iot describe-index \
```

```
    --index-name "AWS_Things"
```

출력:

```
{
    "indexName": "AWS_Things",
    "indexStatus": "ACTIVE",
    "schema": "REGISTRY_AND_SHADOW_AND_CONNECTIVITY_STATUS"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 사물 인덱싱 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeIndex의 섹션을 참조하세요. AWS CLI

### describe-job-execution

다음 코드 예시에서는 describe-job-execution을 사용하는 방법을 보여 줍니다.

AWS CLI

디바이스에서 작업에 대한 실행 세부 정보를 가져오려면

다음 describe-job-execution 예제에서는 지정된 작업에 대한 실행 세부 정보를 가져옵니다.

```
aws iot describe-job-execution \
    --job-id "example-job-01" \
    --thing-name "MyRaspberryPi"
```

출력:

```
{
    "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
    }
```

```
}
```

자세한 내용은 AWS IoT 개발자 안내서의 작업 생성 및 관리(CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeJobExecution의 섹션을 참조하세요. AWS CLI

### describe-job

다음 코드 예시에서는 describe-job을 사용하는 방법을 보여 줍니다.

AWS CLI

작업에 대한 세부 상태를 가져오려면

다음 describe-job 예제에서는 ID가 인 작업에 대한 세부 상태를 가져옵니다example-job-01.

```
aws iot describe-job \
    --job-id "example-job-01"
```

출력:

```
{
    "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": {}
    }
}
```

자세한 내용은 AWS IoT 개발자 안내서의 작업 생성 및 관리(CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeJob의 섹션을 참조하세요. AWS CLI

### describe-mitigation-action

다음 코드 예시에서는 describe-mitigation-action을 사용하는 방법을 보여 줍니다.

AWS CLI

정의된 완화 조치에 대한 세부 정보를 보려면

다음 describe-mitigation-action 예제에서는 지정된 완화 작업에 대한 세부 정보를 표시합니다.

```
aws iot describe-mitigation-action \
    --action-name AddThingsToQuarantineGroupAction
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 DescribeMitigationAction (완화 작업 명령)을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeMitigationAction의 섹션을 참조하세요. AWS CLI

### `describe-provisioning-template-version`

다음 코드 예시에서는 `describe-provisioning-template-version`을 사용하는 방법을 보여 줍니다.

AWS CLI

프로비저닝 템플릿 버전을 설명하려면

다음 `describe-provisioning-template-version` 예제에서는 프로비저닝 템플릿 버전을 설명합니다.

```
aws iot describe-provisioning-template-version \
    --template-name MyTestProvisioningTemplate \
    --version-id 1
```

출력:

```
{
    "versionId": 1,
    "creationDate": 1589308310.574,
    "templateBody": "{
        \"Parameters\":{
            \"SerialNumber\":{
                \"Type\":\"String\"
            },
            \"AWS::IoT::Certificate::Id\":{
                \"Type\":\"String\"
            }
        },
        \"Resources\":{
            \"certificate\":{
                \"Properties\":{
                    \"CertificateId\":{
```

```
                        \"Ref\":\"AWS::IoT::Certificate::Id\"
                    },
                    \"Status\":\"Active\"
                },
                \"Type\":\"AWS::IoT::Certificate\"
            },
            \"policy\":{
                \"Properties\":{
                    \"PolicyName\":\"MyIotPolicy\"
                },
                \"Type\":\"AWS::IoT::Policy\"
            },
            \"thing\":{
                \"OverrideSettings\":{
                    \"AttributePayload\":\"MERGE\",
                    \"ThingGroups\":\"DO_NOTHING\",
                    \"ThingTypeName\":\"REPLACE\"
                },
                \"Properties\":{
                    \"AttributePayload\":{},
                    \"ThingGroups\":[],
                    \"ThingName\":{
                        \"Fn::Join\":[
                            \"\",
                            [
                                \"DemoGroup_\",
                                {\"Ref\":\"SerialNumber\"}
                            ]
                        ]
                    },
                    \"ThingTypeName\":\"VirtualThings\"
                },
                \"Type\":\"AWS::IoT::Thing\"
            }
        }
    }",
    "isDefaultVersion": true
}
```

자세한 내용은 IoT Core 개발자 안내서의 플릿 프로비저닝을 사용하여 디바이스 인증서가 없는 디바이스 프로비저닝을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조DescribeProvisioningTemplateVersion의 섹션을 참조하세요. AWS CLI

### describe-provisioning-template

다음 코드 예시에서는 describe-provisioning-template을 사용하는 방법을 보여 줍니다.

AWS CLI

프로비저닝 템플릿을 설명하려면

다음 describe-provisioning-template 예제에서는 프로비저닝 템플릿을 설명합니다.

```
aws iot describe-provisioning-template \
    --template-name MyTestProvisioningTemplate
```

출력:

```
{
    "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"
}
```

자세한 내용은 IoT Core 개발자 안내서의 플릿 프로비저닝을 사용하여 디바이스 인증서가 없는 디바이스 프로비저닝을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조DescribeProvisioningTemplate의 섹션을 참조하세요. AWS CLI

### `describe-role-alias`

다음 코드 예시에서는 `describe-role-alias`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS IoT 역할 별칭에 대한 정보를 가져오려면

다음 describe-role-alias 예제에서는 지정된 역할 별칭에 대한 세부 정보를 표시합니다.

```
aws iot describe-role-alias \
    --role-alias LightBulbRole
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 IoT 참조DescribeRoleAlias의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조DescribeRoleAlias의 섹션을 참조하세요. AWS CLI

### describe-scheduled-audit

다음 코드 예시에서는 describe-scheduled-audit을 사용하는 방법을 보여 줍니다.

AWS CLI

예약된 감사에 대한 정보를 가져오려면

다음 describe-scheduled-audit 예제에서는 라는 Device Defender 예약 감사에 AWS IOT 대한 자세한 정보를 가져옵니다AWSIoTDeviceDefenderDailyAudit.

```
aws iot describe-scheduled-audit \
    --scheduled-audit-name AWSIoTDeviceDefenderDailyAudit
```

출력:

```
{
```

```
    "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"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 감사 명령을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeScheduledAudit의 섹션을 참조하세요. AWS CLI

### `describe-security-profile`

다음 코드 예시에서는 `describe-security-profile`을 사용하는 방법을 보여 줍니다.

AWS CLI

보안 프로필에 대한 정보를 가져오려면

다음 `describe-security-profile` 예제에서는 라는 AWS IoT Device Defender 보안 프로파일에 대한 정보를 가져옵니다. `PossibleIssue.`

```
aws iot describe-security-profile \
    --security-profile-name PossibleIssue
```

출력:

```
{
    "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
}
```

자세한 내용은 AWS IoT 개발자 안내서의 Detect Commands를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeSecurityProfile의 섹션을 참조하세요. AWS CLI

### `describe-stream`

다음 코드 예시에서는 `describe-stream`을 사용하는 방법을 보여 줍니다.

AWS CLI

스트림에 대한 정보를 가져오려면

다음 `describe-stream` 예제에서는 지정된 스트림에 대한 세부 정보를 표시합니다.

```
aws iot describe-stream \
    --stream-id stream12345
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 IoT 참조DescribeStream의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조DescribeStream의 섹션을 참조하세요. AWS CLI

### `describe-thing-group`

다음 코드 예시에서는 `describe-thing-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

사물 그룹에 대한 정보를 가져오려면

다음 `describe-thing-group` 예제에서는 라는 사물 그룹에 대한 정보를 가져옵니다`HalogenBulbs`.

```
aws iot describe-thing-group \
```

```
    --thing-group-name HalogenBulbs
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 IoT 개발자 안내서의 사물 그룹을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조DescribeThingGroup의 섹션을 참조하세요. AWS CLI

### describe-thing-type

다음 코드 예시에서는 describe-thing-type을 사용하는 방법을 보여 줍니다.

AWS CLI

사물 유형에 대한 정보를 가져오려면

다음 describe-thing-type 예제에서는 AWS 계정에 정의된 지정된 사물 유형에 대한 정보를 표시합니다.

```
aws iot describe-thing-type \
    --thing-type-name "LightBulb"
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 IoT 개발자 안내서의 사물 유형을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조DescribeThingType의 섹션을 참조하세요. AWS CLI

### `describe-thing`

다음 코드 예시에서는 `describe-thing`을 사용하는 방법을 보여 줍니다.

AWS CLI

사물에 대한 세부 정보를 표시하려면

다음 `describe-thing` 예제에서는 AWS 계정의 AWS IoT 레지스트리에 정의된 사물(장치)에 대한 정보를 표시합니다.

aws iot describe-thing --thing-name “MyLightBulb

출력:

```
{
    "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
}
```

자세한 내용은 AWS IoT 개발자 안내서의 레지스트리를 사용하여 사물을 관리하는 방법을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeThing의 섹션을 참조하세요. AWS CLI

### `detach-policy`

다음 코드 예시에서는 `detach-policy`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 사물 그룹에서 AWS IoT 정책을 분리하려면

다음 `detach-policy` 예제에서는 사물 그룹에서 지정된 정책을 분리하고 확장별로 해당 그룹의 모든 사물과 그룹의 하위 그룹에서 지정된 정책을 분리합니다.

```
aws iot detach-policy \
    --target "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs" \
    --policy-name "MyFirstGroup_Core-policy"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IoT 개발자 안내서의 사물 그룹을 참조하세요. AWS IoT

예제 2: 디바이스 인증서에서 AWS IoT 정책을 분리하려면

다음 `detach-policy` 예제에서는 에서 식별한 디바이스 인증서에서 TemperatureSensorPolicy 정책을 분리합니다ARN.

```
aws iot detach-policy \
    --policy-name TemperatureSensorPolicy \
    --target arn:aws:iot:us-
west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DetachPolicy의 섹션을 참조하세요. AWS CLI

### detach-security-profile

다음 코드 예시에서는 detach-security-profile을 사용하는 방법을 보여 줍니다.

AWS CLI

대상에서 보안 프로파일 연결을 해제하려면

다음 detach-security-profile 예제에서는 이름이 인 AWS IoT Device Defender 보안 프로파일Testprofile과 등록된 모든 사물 대상 간의 연결을 제거합니다.

```
aws iot detach-security-profile \
    --security-profile-name Testprofile \
    --security-profile-target-arn "arn:aws:iot:us-west-2:123456789012:all/
registered-things"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 Detect Commands를 참조하세요.

- 자세한 API 내용은 명령 참조DetachSecurityProfile의 섹션을 참조하세요. AWS CLI

### detach-thing-principal

다음 코드 예시에서는 detach-thing-principal을 사용하는 방법을 보여 줍니다.

AWS CLI

인증서/주제를 사물에서 분리하려면

다음 detach-thing-principal 예제에서는 지정된 사물에서 보안 주체를 나타내는 인증서를 제거합니다.

```
aws iot detach-thing-principal \
    --thing-name "MyLightBulb" \
    --principal "arn:aws:iot:us-
west-2:123456789012:cert/604c48437a57b7d5fc5d137c5be75011c6ee67c9a6943683a1acb4b1626bac36"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 레지스트리를 사용하여 사물을 관리하는 방법을 참조하세요.

- 자세한 API 내용은 명령 참조DetachThingPrincipal의 섹션을 참조하세요. AWS CLI

### disable-topic-rule

다음 코드 예시에서는 disable-topic-rule을 사용하는 방법을 보여 줍니다.

AWS CLI

주제 규칙을 비활성화하려면

다음 disable-topic-rule 예제에서는 지정된 주제 규칙을 비활성화합니다.

```
aws iot disable-topic-rule \
    --rule-name "MyPlantPiMoistureAlertRule"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 규칙 보기를 참조하세요.

- 자세한 API 내용은 명령 참조DisableTopicRule의 섹션을 참조하세요. AWS CLI

### enable-topic-rule

다음 코드 예시에서는 enable-topic-rule을 사용하는 방법을 보여 줍니다.

AWS CLI

주제 규칙을 활성화하려면

다음 enable-topic-rule 예제에서는 지정된 주제 규칙을 활성화(또는 다시 활성화)합니다.

```
aws iot enable-topic-rule \
    --rule-name "MyPlantPiMoistureAlertRule"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 규칙 보기를 참조하세요.

- 자세한 API 내용은 명령 참조EnableTopicRule의 섹션을 참조하세요. AWS CLI

### get-behavior-model-training-summaries

다음 코드 예시에서는 get-behavior-model-training-summaries을 사용하는 방법을 보여 줍니다.

AWS CLI

Device Defender의 ML Detect Security Profile 훈련 모델 상태를 나열하려면

다음 get-behavior-model-training-summaries 예제에서는 선택한 보안 프로필의 구성된 동작에 대한 모델 훈련 상태를 나열합니다. 각 동작에 대해 수집된 데이터 포인트의 이름, 모델 상태 및 백분율이 나열됩니다.

```
aws iot get-behavior-model-training-summaries \
    --security-profile-name MySecuirtyProfileName
```

출력:

```
{
    "summaries": [
        {
            "securityProfileName": "MySecuirtyProfileName",
            "behaviorName": "Messages_sent_ML_behavior",
            "modelStatus": "PENDING_BUILD",
            "datapointsCollectionPercentage": 0.0
        },
        {
            "securityProfileName": "MySecuirtyProfileName",
            "behaviorName": "Messages_received_ML_behavior",
            "modelStatus": "PENDING_BUILD",
            "datapointsCollectionPercentage": 0.0
        },
        {
            "securityProfileName": "MySecuirtyProfileName",
            "behaviorName": "Authorization_failures_ML_behavior",
            "modelStatus": "PENDING_BUILD",
            "datapointsCollectionPercentage": 0.0
        },
```

```
        {
            "securityProfileName": "MySecuirtyProfileName",
            "behaviorName": "Message_size_ML_behavior",
            "modelStatus": "PENDING_BUILD",
            "datapointsCollectionPercentage": 0.0
        },
        {
            "securityProfileName": "MySecuirtyProfileName",
            "behaviorName": "Connection_attempts_ML_behavior",
            "modelStatus": "PENDING_BUILD",
            "datapointsCollectionPercentage": 0.0
        },
        {
            "securityProfileName": "MySPNoALerts",
            "behaviorName": "Disconnects_ML_behavior",
            "modelStatus": "PENDING_BUILD",
            "datapointsCollectionPercentage": 0.0
        }
    ]
}
```

자세한 내용은 AWS IoT 개발자 안내서의 GetBehaviorModelTrainingSummaries (Detect Commands)를 참조하세요.

- 자세한 API 내용은 명령 참조GetBehaviorModelTrainingSummaries의 섹션을 참조하세요. AWS CLI

### `get-cardinality`

다음 코드 예시에서는 `get-cardinality`을 사용하는 방법을 보여 줍니다.

AWS CLI

쿼리와 일치하는 고유한 값의 대략적인 수를 반환하려면

다음 설정 스크립트를 사용하여 온도 센서 10개를 나타내는 10개의 사물을 생성할 수 있습니다. 각 새 항목에는 3개의 속성이 있습니다.

```
# 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
```

설정 스크립트의 출력 예:

```
{
    "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"
}
```

다음 `get-cardinality` 예제는 설정 스크립트에서 생성한 10개의 센서를 쿼리하고 온도 센서가 비정상 온도 값을 보고하는 랙 수를 반환합니다. 온도 값이 60 미만이거나 80을 초과하는 경우 온도 센서가 비정상 상태입니다.

```
aws iot get-cardinality \
    --aggregation-field "attributes.rackId" \
    --query-string "thingName:TempSensor* AND attributes.stateNormal:false"
```

출력:

```
{
    "cardinality": 2
}
```

자세한 내용은 AWS IoT 개발자 안내서의 집계 데이터 쿼리<https://docs.aws.amazon.com/iot/latest/developerguide/index-aggregate.html>를 참조하세요.

- 자세한 API 내용은 명령 참조GetCardinality의 섹션을 참조하세요. AWS CLI

### get-effective-policies

다음 코드 예시에서는 get-effective-policies을 사용하는 방법을 보여 줍니다.

AWS CLI

사물에 영향을 미치는 정책을 나열하려면

다음 get-effective-policies 예제에서는 지정된 사물에 영향을 미치는 정책을 나열합니다. 여기에는 속한 모든 그룹에 연결된 정책이 포함됩니다.

```
aws iot get-effective-policies \
    --thing-name TemperatureSensor-001 \
    --principal arn:aws:iot:us-
west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142
```

출력:

```
{
    "effectivePolicies": [
        {
            "policyName": "TemperatureSensorPolicy",
            "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/
TemperatureSensorPolicy",
            "policyDocument": "{
                \"Version\": \"2012-10-17\",
                \"Statement\": [
                    {
                        \"Effect\": \"Allow\",
                        \"Action\": [
                            \"iot:Publish\",
                            \"iot:Receive\"
                        ],
                        \"Resource\": [
                            \"arn:aws:iot:us-west-2:123456789012:topic/topic_1\",
                            \"arn:aws:iot:us-west-2:123456789012:topic/topic_2\"
                        ]
                    },
                    {
                        \"Effect\": \"Allow\",
                        \"Action\": [
                            \"iot:Subscribe\"
                        ],
```

```
                        \"Resource\": [
                            \"arn:aws:iot:us-west-2:123456789012:topicfilter/
topic_1\",
                            \"arn:aws:iot:us-west-2:123456789012:topicfilter/
topic_2\"
                        ]
                    },
                    {
                        \"Effect\": \"Allow\",
                        \"Action\": [
                            \"iot:Connect\"
                        ],
                        \"Resource\": [
                            \"arn:aws:iot:us-west-2:123456789012:client/basicPubSub
\"
                        ]
                    }
                ]
            }"
        }
    ]
}
```

자세한 내용은 IoT 개발자 안내서의 사물에 대한 효과적인 정책 가져오기를 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조GetEffectivePolicies의 섹션을 참조하세요. AWS CLI

### `get-indexing-configuration`

다음 코드 예시에서는 `get-indexing-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

사물 인덱싱 구성을 가져오려면

다음 `get-indexing-configuration` 예제에서는 AWS IoT 플릿 인덱싱에 대한 현재 구성 데이터를 가져옵니다.

```
aws iot get-indexing-configuration
```

출력:

```
{
```

```
    "thingIndexingConfiguration": {
        "thingIndexingMode": "OFF",
        "thingConnectivityIndexingMode": "OFF"
    },
    "thingGroupIndexingConfiguration": {
        "thingGroupIndexingMode": "OFF"
    }
}
```

자세한 내용은 IoT 개발자 안내서의 사물 인덱싱 관리를 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조GetIndexingConfiguration의 섹션을 참조하세요. AWS CLI

### get-job-document

다음 코드 예시에서는 get-job-document을 사용하는 방법을 보여 줍니다.

AWS CLI

작업에 대한 문서를 검색하려면

다음 get-job-document 예제에서는 ID가 인 작업의 문서에 대한 세부 정보를 표시합니다example-job-01.

```
aws iot get-job-document \
    --job-id "example-job-01"
```

출력:

```
{
    "document": "\n{\n    \"operation\":\"customJob\",\n    \"otherInfo\":
\"someValue\"\n}\n"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 작업 생성 및 관리(CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조GetJobDocument의 섹션을 참조하세요. AWS CLI

### get-logging-options

다음 코드 예시에서는 get-logging-options을 사용하는 방법을 보여 줍니다.

AWS CLI

로깅 옵션을 가져오려면

다음 `get-logging-options` 예제에서는 AWS 계정의 현재 로깅 옵션을 가져옵니다.

```
aws iot get-logging-options
```

출력:

```
{
    "roleArn": "arn:aws:iam::123456789012:role/service-role/iotLoggingRole",
    "logLevel": "ERROR"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 제목을 참조하세요.

- 자세한 API 내용은 명령 참조GetLoggingOptions의 섹션을 참조하세요. AWS CLI

### `get-ota-update`

다음 코드 예시에서는 `get-ota-update`을 사용하는 방법을 보여 줍니다.

AWS CLI

OTA 업데이트에 대한 정보를 검색하려면

다음 `get-ota-update` 예제에서는 지정된 OTA 업데이트에 대한 세부 정보를 표시합니다.

```
aws iot get-ota-update \
    --ota-update-id ota12345
```

출력:

```
{
    "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": {
        }
    }
}
```

자세한 내용은 AWS IoT API 참조의 GetOTAUpdate를 참조하세요.

- 자세한 API 내용은 명령 참조GetOtaUpdate의 섹션을 참조하세요. AWS CLI

### get-percentiles

다음 코드 예시에서는 get-percentiles을 사용하는 방법을 보여 줍니다.

AWS CLI

쿼리와 일치하는 집계된 값을 백분위수 그룹으로 그룹화하려면

다음 설정 스크립트를 사용하여 온도 센서 10개를 나타내는 10개의 사물을 생성할 수 있습니다. 각 새 객체에는 속성이 1개 있습니다.

```
# 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
```

설정 스크립트의 출력 예:

```
{
    "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"
}
```

다음 `get-percentiles` 예제는 설정 스크립트에서 생성한 10개의 센서를 쿼리하고 지정된 각 백분위수 그룹에 대한 값을 반환합니다. 백분위수 그룹 “10”에는 쿼리와 일치하는 값의 약 10%에서 발생하는 집계된 필드 값이 포함됩니다. 다음 출력에서 {'백분율': 10.0, '값': 67.7}은 온도 값의 약 10.0%가 67.7 미만임을 의미합니다.

```
aws iot get-percentiles \
    --aggregation-field "attributes.temperature" \
    --query-string "thingName:TempSensor*" \
    --percents 10 25 50 75 90
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 AWS IoT 개발자 안내서의 집계 데이터 쿼리를 참조하세요.

- 자세한 API 내용은 명령 참조GetPercentiles의 섹션을 참조하세요. AWS CLI

### get-policy-version

다음 코드 예시에서는 get-policy-version을 사용하는 방법을 보여 줍니다.

AWS CLI

특정 버전의 정책에 대한 정보를 가져오려면

다음 get-policy-version 예제에서는 지정된 정책의 첫 번째 버전에 대한 정보를 가져옵니다.

```
aws iot get-policy \
    --policy-name UpdateDeviceCertPolicy
    --policy-version-id "1"
```

출력:

```
{
```

```
    "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"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 IoT 정책을 AWS 참조하세요 IoT.

- 자세한 API 내용은 명령 참조GetPolicyVersion의 섹션을 참조하세요. AWS CLI

### `get-policy`

다음 코드 예시에서는 `get-policy`을 사용하는 방법을 보여 줍니다.

AWS CLI

정책의 기본 버전에 대한 정보를 가져오려면

다음 `get-policy` 예제에서는 지정된 정책의 기본 버전에 대한 정보를 검색합니다.

```
aws iot get-policy \
    --policy-name UpdateDeviceCertPolicy
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 IoT 정책을 AWS 참조하세요 IoT.

- 자세한 API 내용은 명령 참조GetPolicy의 섹션을 참조하세요. AWS CLI

### get-registration-code

다음 코드 예시에서는 get-registration-code을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 계정별 등록 코드를 가져오려면

다음 get-registration-code 예제에서는 AWS 계정별 등록 코드를 검색합니다.

```
aws iot get-registration-code
```

출력:

```
{
    "registrationCode":
 "15c51ae5e36ba59ba77042df1115862076bea4bd15841c838fcb68d5010a614c"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 자체 인증서 사용을 참조하세요.

- 자세한 API 내용은 명령 참조GetRegistrationCode의 섹션을 참조하세요. AWS CLI

### get-statistics

다음 코드 예시에서는 get-statistics을 사용하는 방법을 보여 줍니다.

AWS CLI

디바이스 인덱스에서 집계 데이터를 검색하려면

다음 get-statistics 예제는 디바이스 섀도우에서 로 connectivity.connected 설정된 속성false(즉, 연결되지 않은 디바이스 수)이 있는 사물 수를 반환합니다.

```
aws iot get-statistics \
    --index-name AWS_Things \
```

```
    --query-string "connectivity.connected:false"
```

출력:

```
{
    "statistics": {
        "count": 6
    }
}
```

자세한 내용은 AWS IoT 개발자 안내서의 디바이스 플릿에 대한 통계 가져오기를 참조하세요.

- 자세한 API 내용은 명령 참조GetStatistics의 섹션을 참조하세요. AWS CLI

### get-topic-rule-destination

다음 코드 예시에서는 get-topic-rule-destination을 사용하는 방법을 보여 줍니다.

AWS CLI

주제 규칙 대상을 가져오려면

다음 get-topic-rule-destination 예제에서는 주제 규칙 대상에 대한 정보를 가져옵니다.

```
aws iot get-topic-rule-destination \
    --arn "arn:aws:iot:us-west-2:123456789012:ruledestination/http/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
```

출력:

```
{
    "topicRuleDestination": {
        "arn": "arn:aws:iot:us-west-2:123456789012:ruledestination/http/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
        "status": "DISABLED",
        "httpUrlProperties": {
            "confirmationUrl": "https://example.com"
        }
    }
}
```

자세한 내용은 AWS IoT 개발자 안내서의 주제 규칙 대상 작업을 참조하세요.

- 자세한 API 내용은 명령 참조GetTopicRuleDestination의 섹션을 참조하세요. AWS CLI

### get-topic-rule

다음 코드 예시에서는 get-topic-rule을 사용하는 방법을 보여 줍니다.

AWS CLI

규칙에 대한 정보를 가져오려면

다음 get-topic-rule 예제에서는 지정된 규칙에 대한 정보를 가져옵니다.

```
aws iot get-topic-rule \
    --rule-name MyRPiLowMoistureAlertRule
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 IoT 개발자 안내서의 규칙 보기를 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조GetTopicRule의 섹션을 참조하세요. AWS CLI

### get-v2-logging-options

다음 코드 예시에서는 get-v2-logging-options을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 로깅 옵션을 나열하려면

다음 get-v2-logging-options 예제에서는 AWS IoT 에 대한 현재 로깅 옵션을 나열합니다.

```
aws iot get-v2-logging-options
```

출력:

```
{
    "roleArn": "arn:aws:iam::094249569039:role/service-role/iotLoggingRole",
    "defaultLogLevel": "WARN",
    "disableAllLogs": false
}
```

자세한 내용은 AWS IoT 개발자 안내서의 제목을 참조하세요.

- API 자세한 내용은 AWS CLI 명령 참조의 GetV2LoggingOptions를 참조하세요.

### list-active-violations

다음 코드 예시에서는 list-active-violations을 사용하는 방법을 보여 줍니다.

AWS CLI

활성 위반을 나열하려면

다음 list-active-violations 예제에서는 지정된 보안 프로파일에 대한 모든 위반을 나열합니다.

```
aws iot list-active-violations \
    --security-profile-name Testprofile
```

출력:


```
{
    "activeViolations": [
        {
            "violationId": "174db59167fa474c80a652ad1583fd44",
            "thingName": "iotconsole-1560269126751-1",
            "securityProfileName": "Testprofile",
            "behavior": {
                "name": "Authorization",
                "metric": "aws:num-authorization-failures",
                "criteria": {
                    "comparisonOperator": "greater-than",
                    "value": {
                        "count": 10
                    },
                    "durationSeconds": 300,
                    "consecutiveDatapointsToAlarm": 1,
                    "consecutiveDatapointsToClear": 1
                }
            },
            "lastViolationValue": {
                "count": 0
            },
            "lastViolationTime": 1560293700.0,
            "violationStartTime": 1560279000.0
        },
        {
            "violationId": "c8a9466a093d3b7b35cd44ca58bdbeab",
            "thingName": "TvnQoEoU",
            "securityProfileName": "Testprofile",
            "behavior": {
                "name": "CellularBandwidth",
                "metric": "aws:message-byte-size",
                "criteria": {
                    "comparisonOperator": "greater-than",
                    "value": {
                        "count": 128
                    },
                    "consecutiveDatapointsToAlarm": 1,
                    "consecutiveDatapointsToClear": 1
                }
            },
            "lastViolationValue": {
```

```
                "count": 110
            },
            "lastViolationTime": 1560369000.0,
            "violationStartTime": 1560276600.0
        },
        {
            "violationId": "74aa393adea02e6648f3ac362beed55e",
            "thingName": "iotconsole-1560269232412-2",
            "securityProfileName": "Testprofile",
            "behavior": {
                "name": "Authorization",
                "metric": "aws:num-authorization-failures",
                "criteria": {
                    "comparisonOperator": "greater-than",
                    "value": {
                        "count": 10
                    },
                    "durationSeconds": 300,
                    "consecutiveDatapointsToAlarm": 1,
                    "consecutiveDatapointsToClear": 1
                }
            },
            "lastViolationValue": {
                "count": 0
            },
            "lastViolationTime": 1560276600.0,
            "violationStartTime": 1560276600.0
        },
        {
            "violationId": "1e6ab5f7cf39a1466fcd154e1377e406",
            "thingName": "TvnQoEoU",
            "securityProfileName": "Testprofile",
            "behavior": {
                "name": "Authorization",
                "metric": "aws:num-authorization-failures",
                "criteria": {
                    "comparisonOperator": "greater-than",
                    "value": {
                        "count": 10
                    },
                    "durationSeconds": 300,
                    "consecutiveDatapointsToAlarm": 1,
                    "consecutiveDatapointsToClear": 1
                }
```

```
            },
            "lastViolationValue": {
                "count": 0
            },
            "lastViolationTime": 1560369000.0,
            "violationStartTime": 1560276600.0
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListActiveViolations의 섹션을 참조하세요. AWS CLI

### `list-attached-policies`

다음 코드 예시에서는 `list-attached-policies`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 그룹에 연결된 정책을 나열하려면

다음 `list-attached-policies` 예제에서는 지정된 그룹에 연결된 정책을 나열합니다.

```
aws iot list-attached-policies \
    --target "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs"
```

출력:

```
{
    "policies": [
        {
            "policyName": "UpdateDeviceCertPolicy",
            "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/
UpdateDeviceCertPolicy"
        }
    ]
}
```

자세한 내용은 IoT 개발자 안내서의 사물 그룹을 참조하세요. AWS IoT

예제 2: 디바이스 인증서에 연결된 정책을 나열하려면

다음 `list-attached-policies` 예제에서는 디바이스 인증서에 연결된 AWS IoT 정책을 나열합니다. 인증서는 로 식별됩니다ARN.

```
aws iot list-attached-policies \
    --target arn:aws:iot:us-
west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142
```

출력:

```
{
    "policies": [
        {
            "policyName": "TemperatureSensorPolicy",
            "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/
TemperatureSensorPolicy"
        }
    ]
}
```

자세한 내용은 IoT 개발자 안내서의 사물 그룹을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조ListAttachedPolicies의 섹션을 참조하세요. AWS CLI

### `list-audit-findings`

다음 코드 예시에서는 `list-audit-findings`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 감사의 모든 결과를 나열하려면

다음 `list-audit-findings` 예제에서는 지정된 작업 ID를 사용하여 AWS IoT Device Defender 감사의 모든 결과를 나열합니다.

```
aws iot list-audit-findings \
    --task-id a3aea009955e501a31b764abe1bebd3d
```

출력:

```
{
    "findings": []
```

```
}
```

예제 2: 감사 확인 유형에 대한 조사 결과를 나열하려면

다음 `list-audit-findings` 예제는 디바이스가 디바이스 인증서를 공유하는 2019년 6월 5일부터 2019년 6월 19일까지 실행된 AWS IoT Device Defender 감사 결과를 보여줍니다. 수표 이름을 지정할 때는 시작 및 종료 시간을 제공해야 합니다.

```
aws iot list-audit-findings \
    --check-name DEVICE_CERTIFICATE_SHARED_CHECK \
    --start-time 1559747125 \
    --end-time 1560962028
```

출력:

```
{
    "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": "ZipxgAIl"
                    },
                    "additionalInfo": {
                        "CONNECTION_TIME": "1560086374068"
                    }
                },
                {
                    "resourceType": "CLIENT_ID",
                    "resourceIdentifier": {
```

```
                        "clientId": "ZipxgAIl"
                    },
                    "additionalInfo": {
                        "CONNECTION_TIME": "1560081552187",
                        "DISCONNECTION_TIME": "1560086371552"
                    }
                },
                {
                    "resourceType": "CLIENT_ID",
                    "resourceIdentifier": {
                        "clientId": "ZipxgAIl"
                    },
                    "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"
        }
    ]
}
```

자세한 내용은 AWS IoT 개발자 안내서의 감사 명령을 참조하세요.

- 자세한 API 내용은 명령 참조ListAuditFindings의 섹션을 참조하세요. AWS CLI

### `list-audit-mitigation-actions-executions`

다음 코드 예시에서는 `list-audit-mitigation-actions-executions`을 사용하는 방법을 보여 줍니다.

AWS CLI

감사 완화 작업 실행의 세부 정보를 나열하려면

감사 완화 조치 태스크는 AWS IoT Device Defender 감사의 하나 이상의 결과에 완화 조치를 적용합니다. 다음 `list-audit-mitigation-actions-executions` 예제에서는 가 지정된 완화 작업 작업 `taskId` 및 가 지정된 결과에 대한 세부 정보를 나열합니다.

```
aws iot list-audit-mitigation-actions-executions \
    --task-id myActionsTaskId \
    --finding-id 0edbaaec-2fe1-4cf5-abc9-d4c3e51f7464
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS IoT 개발자 안내서의 ListAuditMitigationActionsExecutions (완화 작업 명령)을 참조하세요.

- 자세한 API 내용은 명령 참조ListAuditMitigationActionsExecutions의 섹션을 참조하세요. AWS CLI

### `list-audit-mitigation-actions-tasks`

다음 코드 예시에서는 `list-audit-mitigation-actions-tasks`을 사용하는 방법을 보여 줍니다.

AWS CLI

감사 완화 조치 작업을 나열하려면

다음 `list-audit-mitigation-actions-tasks` 예제에서는 지정된 기간 내에 결과에 적용된 완화 조치를 나열합니다.

```
aws iot list-audit-mitigation-actions-tasks \
    --start-time 1594157400 \
    --end-time 1594157430
```

출력:

```
{
    "tasks": [
        {
                "taskId": "0062f2d6-3999-488f-88c7-bef005414103",
                "startTime": "2020-07-07T14:30:15.172000-07:00",
            "taskStatus": "COMPLETED"
        }
    ]
}
```

자세한 내용은 AWS IoT 개발자 안내서의 ListAuditMitigationActionsTasks (완화 작업 명령)을 참조하세요.

- 자세한 API 내용은 명령 참조ListAuditMitigationActionsTasks의 섹션을 참조하세요. AWS CLI

### `list-audit-suppressions`

다음 코드 예시에서는 `list-audit-suppressions`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 감사 결과 금지를 나열하려면

다음 `list-audit-suppressions` 예제에서는 모든 활성 감사 결과 억제를 나열합니다.

```
aws iot list-audit-suppressions
```

출력:

```
{
    "suppressions": [
        {
        "checkName": "DEVICE_CERTIFICATE_EXPIRING_CHECK",
            "resourceIdentifier": {
                "deviceCertificateId": "c7691e<shortened>"
            },
        "expirationDate": 1597881600.0,
        "suppressIndefinitely": false
        }
    ]
}
```

자세한 내용은 IoT 개발자 안내서의 감사 결과 금지를 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조ListAuditSuppressions의 섹션을 참조하세요. AWS CLI

### `list-audit-tasks`

다음 코드 예시에서는 `list-audit-tasks`을 사용하는 방법을 보여 줍니다.

AWS CLI

감사의 모든 결과를 나열하려면

다음 `list-audit-tasks` 예제에서는 2019년 6월 5일부터 2019년 6월 12일까지 실행된 감사 작업을 나열합니다.

```
aws iot list-audit-tasks \
    --start-time 1559747125 \
    --end-time 1560357228
```

출력:

```
{
    "tasks": [
        {
            "taskId": "a3aea009955e501a31b764abe1bebd3d",
            "taskStatus": "COMPLETED",
            "taskType": "ON_DEMAND_AUDIT_TASK"
        },
        {
            "taskId": "f76b4b5102b632cd9ae38a279c266da1",
            "taskStatus": "COMPLETED",
            "taskType": "SCHEDULED_AUDIT_TASK"
        },
        {
            "taskId": "51d9967d9f9ff4d26529505f6d2c444a",
            "taskStatus": "COMPLETED",
            "taskType": "SCHEDULED_AUDIT_TASK"
        },
        {
            "taskId": "eeef61068b0eb03c456d746c5a26ee04",
            "taskStatus": "COMPLETED",
            "taskType": "SCHEDULED_AUDIT_TASK"
        },
        {
            "taskId": "041c49557b7c7b04c079a49514b55589",
            "taskStatus": "COMPLETED",
            "taskType": "SCHEDULED_AUDIT_TASK"
        },
        {
            "taskId": "82c7f2afac1562d18a4560be73998acc",
            "taskStatus": "COMPLETED",
            "taskType": "SCHEDULED_AUDIT_TASK"
        },
        {
            "taskId": "bade6b5efd2e1b1569822f6021b39cf5",
```

```
            "taskStatus": "COMPLETED",
            "taskType": "SCHEDULED_AUDIT_TASK"
        },
        {
            "taskId": "c23f6233ba2d35879c4bb2810fb5ffd6",
            "taskStatus": "COMPLETED",
            "taskType": "SCHEDULED_AUDIT_TASK"
        },
        {
            "taskId": "ac9086b7222a2f5e2e17bb6fd30b3aeb",
            "taskStatus": "COMPLETED",
            "taskType": "SCHEDULED_AUDIT_TASK"
        }
    ]
}
```

자세한 내용은 AWS IoT 개발자 안내서의 감사 명령을 참조하세요.

- 자세한 API 내용은 명령 참조ListAuditTasks의 섹션을 참조하세요. AWS CLI

### `list-authorizers`

다음 코드 예시에서는 `list-authorizers`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 권한 부여자를 나열하려면

다음 `list-authorizers` 예제에서는 AWS 계정의 사용자 지정 권한 부여자를 나열합니다.

```
aws iot list-authorizers
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 IoT 참조ListAuthorizers의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조ListAuthorizers의 섹션을 참조하세요. AWS CLI

### `list-billing-groups`

다음 코드 예시에서는 `list-billing-groups`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 계정 및 리전의 결제 그룹을 나열하려면

다음 `list-billing-groups` 예제에서는 AWS 계정 및 AWS 리전에 정의된 모든 결제 그룹을 나열합니다.

```
aws iot list-billing-groups
```

출력:

```
{
    "billingGroups": [
        {
            "groupName": "GroupOne",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:billinggroup/GroupOne"
        }
    ]
}
```

자세한 내용은 IoT 개발자 안내서의 결제 그룹을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조ListBillingGroups의 섹션을 참조하세요. AWS CLI

### list-ca-certificates

다음 코드 예시에서는 list-ca-certificates을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 계정에 등록된 CA 인증서를 나열하려면

다음 list-ca-certificates 예제에서는 AWS 계정에 등록된 CA 인증서를 나열합니다.

```
aws iot list-ca-certificates
```

출력:

```
{
    "certificates": [
        {
            "certificateArn": "arn:aws:iot:us-west-2:123456789012:cacert/
f4efed62c0142f16af278166f61962501165c4f053629520742646005 8cd1467",
            "certificateId":
 "f4efed62c0142f16af278166f61962501165c4f053629520742646005 8cd1467",
            "status": "INACTIVE",
            "creationDate": 1569365372.053
        }
    ]
}
```

자세한 내용은 AWS IoT 개발자 안내서의 자체 인증서 사용을 참조하세요.

- 자세한 API 내용은 명령 참조ListCaCertificates의 섹션을 참조하세요. AWS CLI

### list-certificates-by-ca

다음 코드 예시에서는 list-certificates-by-ca을 사용하는 방법을 보여 줍니다.

AWS CLI

CA 인증서로 서명된 모든 디바이스 인증서를 나열하려면

다음 list-certificates-by-ca 예제에서는 지정된 CA 인증서로 서명된 AWS 계정의 모든 디바이스 인증서를 나열합니다.

```
aws iot list-certificates-by-ca \
```

```
    --ca-certificate-
id f4efed62c0142f16af278166f61962501165c4f053629520742646005 8cd1467
```

출력:

```
{
    "certificates": [
        {
            "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",
            "certificateId":
 "488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",
            "status": "ACTIVE",
            "creationDate": 1569363250.557
        }
    ]
}
```

자세한 내용은 AWS IoT API 참조의 ListCertificatesByCA를 참조하세요.

- 자세한 API 내용은 명령 참조ListCertificatesByCa의 섹션을 참조하세요. AWS CLI

### list-certificates

다음 코드 예시에서는 list-certificates을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: AWS 계정에 등록된 인증서를 나열하려면

다음 list-certificates 예제에서는 계정에 등록된 모든 인증서를 나열합니다. 기본 페이징 제한인 25보다 많은 경우 이 명령의 nextMarker 응답 값을 사용하여 다음 명령에 제공하여 다음 결과 배치를 가져올 수 있습니다. 값 없이 가 nextMarker 반환될 때까지 반복합니다.

```
aws iot list-certificates
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListCertificates의 섹션을 참조하세요. AWS CLI

### `list-custom-metrics`

다음 코드 예시에서는 `list-custom-metrics`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 지표를 나열하려면

다음 `list-custom-metrics` 예제에서는 모든 사용자 지정 지표를 나열합니다.

```
aws iot list-custom-metrics \
    --region us-east-1
```

출력:

```
{
    "metricNames": [
        "batteryPercentage"
    ]
}
```

자세한 내용은 AWS IoT Core 개발자 안내서의 사용자 지정 지표를 참조하세요.

- 자세한 API 내용은 명령 참조ListCustomMetrics의 섹션을 참조하세요. AWS CLI

### `list-dimensions`

다음 코드 예시에서는 `list-dimensions`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 계정의 차원을 나열하려면

다음 `list-dimensions` 예제에서는 AWS 계정에 정의된 모든 AWS IoT Device Defender 차원을 나열합니다.

```
aws iot list-dimensions
```

출력:

```
{
    "dimensionNames": [
        "TopicFilterForAuthMessages",
        "TopicFilterForActivityMessages"
    ]
}
```

자세한 내용은 AWS IoT 개발자 안내서의 Detect Commands를 참조하세요.

- 자세한 API 내용은 명령 참조ListDimensions의 섹션을 참조하세요. AWS CLI

### `list-domain-configurations`

다음 코드 예시에서는 `list-domain-configurations`을 사용하는 방법을 보여 줍니다.

AWS CLI

도메인 구성을 나열하려면

다음 `list-domain-configurations` 예제에서는 지정된 서비스 유형이 있는 AWS 계정의 도메인 구성을 나열합니다.

```
aws iot list-domain-configurations \
    --service-type "DATA"
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS IoT 개발자 안내서의 구성 가능한 엔드포인트를 참조하세요.

- 자세한 API 내용은 명령 참조ListDomainConfigurations의 섹션을 참조하세요. AWS CLI

### `list-indices`

다음 코드 예시에서는 `list-indices`을 사용하는 방법을 보여 줍니다.

AWS CLI

구성된 검색 인덱스를 나열하려면

다음 `list-indices` 예제에서는 AWS 계정에 구성된 모든 검색 인덱스를 나열합니다. 사물 인덱싱을 활성화하지 않은 경우 인덱스가 없을 수 있습니다.

```
aws iot list-indices
```

출력:

```
{
```

```
    "indexNames": [
        "AWS_Things"
    ]
}
```

자세한 내용은 AWS IoT 개발자 안내서의 사물 인덱싱 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ListIndices의 섹션을 참조하세요. AWS CLI

### `list-job-executions-for-job`

다음 코드 예시에서는 `list-job-executions-for-job`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 계정의 작업을 나열하려면

다음 `list-job-executions-for-job` 예제에서는 에서 지정한 AWS 계정의 작업에 대한 모든 작업 실행을 나열합니다jobId.

```
aws iot list-job-executions-for-job \
    --job-id my-ota-job
```

출력:

```
{
     "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
             }
         }
     ]
 }
```

자세한 내용은 AWS IoT 개발자 안내서의 작업 생성 및 관리(CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조ListJobExecutionsForJob의 섹션을 참조하세요. AWS CLI

### list-job-executions-for-thing

다음 코드 예시에서는 list-job-executions-for-thing을 사용하는 방법을 보여 줍니다.

AWS CLI

사물에 대해 실행된 작업을 나열하려면

다음 list-job-executions-for-thing 예제에서는 라는 이름의 사물에 대해 실행된 모든 작업을 나열합니다MyRaspberryPi.

```
aws iot list-job-executions-for-thing \
    --thing-name "MyRaspberryPi"
```

출력:

```
{
    "executionSummaries": [
        {
            "jobId": "example-job-01",
            "jobExecutionSummary": {
                "status": "QUEUED",
                "queuedAt": 1560787023.636,
                "lastUpdatedAt": 1560787023.636,
                "executionNumber": 1
            }
        }
    ]
}
```

자세한 내용은 AWS IoT 개발자 안내서의 작업 생성 및 관리(CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조ListJobExecutionsForThing의 섹션을 참조하세요. AWS CLI

### list-jobs

다음 코드 예시에서는 list-jobs을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 계정의 작업을 나열하려면

다음 list-jobs 예제에서는 작업 상태별로 정렬된 AWS 계정의 모든 작업을 나열합니다.

```
aws iot list-jobs
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 AWS IoT 개발자 안내서의 작업 생성 및 관리(CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조ListJobs의 섹션을 참조하세요. AWS CLI

### `list-mitigation-actions`

다음 코드 예시에서는 `list-mitigation-actions`을 사용하는 방법을 보여 줍니다.

AWS CLI

정의된 모든 완화 조치를 나열하려면

다음 `list-mitigation-actions` 예제에서는 AWS 계정 및 리전에 대해 정의된 모든 완화 조치를 나열합니다. 각 작업에 대해 이름, ARN및 생성 날짜가 나열됩니다.

```
aws iot list-mitigation-actions
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS IoT 개발자 안내서의 ListMitigationActions (완화 작업 명령)을 참조하세요.

- 자세한 API 내용은 명령 참조ListMitigationActions의 섹션을 참조하세요. AWS CLI

### `list-mitigations-actions`

다음 코드 예시에서는 `list-mitigations-actions`을 사용하는 방법을 보여 줍니다.

AWS CLI

정의된 모든 완화 조치를 나열하려면

다음 `list-mitigations-actions` 예제에서는 AWS 계정 및 리전에 대해 정의된 모든 완화 조치를 나열합니다. 각 작업에 대해 이름, ARN및 생성 날짜가 나열됩니다.

```
aws iot list-mitigation-actions
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS IoT 개발자 안내서의 ListMitigationActions (완화 작업 명령)을 참조하세요.

- 자세한 API 내용은 명령 참조ListMitigationsActions의 섹션을 참조하세요. AWS CLI

### `list-ota-updates`

다음 코드 예시에서는 `list-ota-updates`을 사용하는 방법을 보여 줍니다.

AWS CLI

계정에 대한 OTA 업데이트를 나열하려면

다음 `list-ota-updates` 예제에서는 사용 가능한 OTA 업데이트를 나열합니다.

```
aws iot list-ota-updates
```

출력:

```
{
    "otaUpdates": [
        {
            "otaUpdateId": "itsaupdate",
            "otaUpdateArn": "arn:aws:iot:us-west-2:123456789012:otaupdate/
itsaupdate",
            "creationDate": 1557863215.995
        }
    ]
}
```

자세한 내용은 AWS IoT API 참조의 ListOTAUpdates을 참조하세요.

- 자세한 API 내용은 명령 참조ListOtaUpdates의 섹션을 참조하세요. AWS CLI

### list-outgoing-certificates

다음 코드 예시에서는 list-outgoing-certificates을 사용하는 방법을 보여 줍니다.

AWS CLI

다른 AWS 계정으로 전송되는 인증서를 나열하려면

다음 list-outgoing-certificates 예제에서는 transfer-certificate 명령을 사용하여 다른 AWS 계정으로 전송 중인 모든 디바이스 인증서를 나열합니다.

```
aws iot list-outgoing-certificates
```

출력:

```
{
    "outgoingCertificates": [
        {
            "certificateArn": "arn:aws:iot:us-
west-2:030714055129:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",
            "certificateId":
 "488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",
            "transferredTo": "030714055129",
            "transferDate": 1569427780.441,
            "creationDate": 1569363250.557
        }
    ]
}
```

자세한 내용은 IoT 참조ListOutgoingCertificates의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조ListOutgoingCertificates의 섹션을 참조하세요. AWS CLI

### list-policies

다음 코드 예시에서는 list-policies을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 계정에 정의된 정책을 나열하려면

다음 list-policies 예제에서는 AWS 계정에 정의된 모든 정책을 나열합니다.

```
aws iot list-policies
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS IoT 개발자 안내서의 IoT 정책을 AWS 참조하세요 IoT.

- 자세한 API 내용은 명령 참조ListPolicies의 섹션을 참조하세요. AWS CLI

### `list-policy-versions`

다음 코드 예시에서는 `list-policy-versions`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 정책의 모든 버전을 보려면

다음 `list-policy-versions` 예제에서는 지정된 정책의 모든 버전과 생성 날짜를 나열합니다.

```
aws iot list-policy-versions \
    --policy-name LightBulbPolicy
```

출력:

```
{
    "policyVersions": [
        {
            "versionId": "2",
            "isDefaultVersion": true,
            "createDate": 1559925941.924
        },
        {
            "versionId": "1",
            "isDefaultVersion": false,
            "createDate": 1559925941.924
        }
    ]
}
```

자세한 내용은 AWS IoT 개발자 안내서의 IoT 정책을 AWS 참조하세요 IoT.

- 자세한 API 내용은 명령 참조ListPolicyVersions의 섹션을 참조하세요. AWS CLI

### `list-principal-things`

다음 코드 예시에서는 `list-principal-things`을 사용하는 방법을 보여 줍니다.

AWS CLI

보안 주체와 연결된 사물을 나열하려면

다음 `list-principal-things` 예제에서는 에서 지정한 보안 주체에 연결된 사물을 나열합니다 ARN.

```
aws iot list-principal-things \
    --principal arn:aws:iot:us-
west-2:123456789012:cert/2e1eb273792174ec2b9bf4e9b37e6c6c692345499506002a351597670552 78e8
```

출력:

```
{
    "things": [
        "DeskLamp",
        "TableLamp"
    ]
```

```
}
```

자세한 내용은 IoT 참조ListPrincipalThings의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조ListPrincipalThings의 섹션을 참조하세요. AWS CLI

### `list-provisioning-template-versions`

다음 코드 예시에서는 `list-provisioning-template-versions`을 사용하는 방법을 보여 줍니다.

AWS CLI

프로비저닝 템플릿 버전을 나열하려면

다음 `list-provisioning-template-versions` 예제에서는 지정된 프로비저닝 템플릿의 사용 가능한 버전을 나열합니다.

```
aws iot list-provisioning-template-versions \
    --template-name "widget-template"
```

출력:

```
{
    "versions": [
        {
            "versionId": 1,
            "creationDate": 1574800471.339,
            "isDefaultVersion": true
        },
        {
            "versionId": 2,
            "creationDate": 1574801192.317,
            "isDefaultVersion": false
        }
    ]
}
```

자세한 내용은 AWS IoT Core 개발자 안내서의 IoT 보안 터널링을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조ListProvisioningTemplateVersions의 섹션을 참조하세요. AWS CLI

### list-provisioning-templates

다음 코드 예시에서는 list-provisioning-templates을 사용하는 방법을 보여 줍니다.

AWS CLI

프로비저닝 템플릿을 나열하려면

다음 list-provisioning-templates 예제에서는 AWS 계정의 모든 프로비저닝 템플릿을 나열합니다.

```
aws iot list-provisioning-templates
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 AWS IoT Core 개발자 안내서의 IoT 보안 터널링을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조ListProvisioningTemplates의 섹션을 참조하세요. AWS CLI

### list-role-aliases

다음 코드 예시에서는 list-role-aliases을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 계정의 AWS IoT 역할 별칭을 나열하려면

다음 list-role-aliases 예제에서는 AWS 계정의 AWS IoT 역할 별칭을 나열합니다.

```
aws iot list-role-aliases
```

출력:

```
{
    "roleAliases": [
        "ResidentAlias",
        "ElectricianAlias"
    ]
}
```

자세한 내용은 IoT 참조ListRoleAliases의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조ListRoleAliases의 섹션을 참조하세요. AWS CLI

### `list-scheduled-audits`

다음 코드 예시에서는 `list-scheduled-audits`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 계정에 대해 예약된 감사를 나열하려면

다음 `list-scheduled-audits` 예제에서는 AWS 계정에 예약된 모든 감사를 나열합니다.

```
aws iot list-scheduled-audits
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS IoT 개발자 안내서의 감사 명령을 참조하세요.

- 자세한 API 내용은 명령 참조ListScheduledAudits의 섹션을 참조하세요. AWS CLI

### `list-security-profiles-for-target`

다음 코드 예시에서는 `list-security-profiles-for-target`을 사용하는 방법을 보여 줍니다.

AWS CLI

대상에 연결된 보안 프로필을 나열하려면

다음 `list-security-profiles-for-target` 예제에서는 등록되지 않은 디바이스에 연결된 AWS IoT Device Defender 보안 프로필을 나열합니다.

```
aws iot list-security-profiles-for-target  \
    --security-profile-target-arn "arn:aws:iot:us-west-2:123456789012:all/
unregistered-things"
```

출력:

```
{
    "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"
            }
        }
    ]
}
```

자세한 내용은 AWS IoT 개발자 안내서의 Detect Commands를 참조하세요.

- 자세한 API 내용은 명령 참조ListSecurityProfilesForTarget의 섹션을 참조하세요. AWS CLI

### `list-security-profiles`

다음 코드 예시에서는 `list-security-profiles`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 계정의 보안 프로필을 나열하려면

다음 `list-security-profiles` 예제에서는 AWS 계정에 정의된 모든 AWS IoT Device Defender 보안 프로필을 나열합니다.

```
aws iot list-security-profiles
```

출력:

```
{
    "securityProfileIdentifiers": [
        {
            "name": "Testprofile",
            "arn": "arn:aws:iot:us-west-2:123456789012:securityprofile/Testprofile"
        }
    ]
}
```

자세한 내용은 AWS IoT 개발자 안내서의 Detect Commands를 참조하세요.

- 자세한 API 내용은 명령 참조ListSecurityProfiles의 섹션을 참조하세요. AWS CLI

### `list-streams`

다음 코드 예시에서는 `list-streams`을 사용하는 방법을 보여 줍니다.

AWS CLI

계정의 스트림을 나열하려면

다음 `list-streams` 예제에서는 AWS 계정의 모든 스트림을 나열합니다.

```
aws iot list-streams
```

출력:

```
{
    "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."
        }
    ]
}
```

자세한 내용은 IoT 참조ListStreams의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조ListStreams의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스와 연결된 태그 및 해당 값을 표시하려면

다음 `list-tags-for-resource` 예제에서는 사물 그룹 과 연결된 태그 및 값을 표시합니다`LightBulbs`.

```
aws iot list-tags-for-resource \
    --resource-arn "arn:aws:iot:us-west-2:094249569039:thinggroup/LightBulbs"
```

출력:

```
{
    "tags": [
        {
            "Key": "Assembly",
            "Value": "Fact1NW"
        },
        {
            "Key": "MyTag",
            "Value": "777"
        }
    ]
}
```

자세한 내용은 AWS IoT 개발자 안내서의 IoT 리소스 태그 지정을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### `list-targets-for-policy`

다음 코드 예시에서는 `list-targets-for-policy`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS IoT 정책과 연결된 보안 주체를 나열하려면

다음 `list-targets-for-policy` 예제에서는 지정된 정책이 연결된 디바이스 인증서를 나열합니다.

```
aws iot list-targets-for-policy \
    --policy-name UpdateDeviceCertPolicy
```

출력:

```
{
    "targets": [
        "arn:aws:iot:us-
west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",
        "arn:aws:iot:us-west-2:123456789012:cert/
d1eb269fb55a628552143c8f96eb3c258fcd5331ea113e766ba0c82bf225f0be"
    ]
}
```

자세한 내용은 IoT 개발자 안내서의 사물 그룹을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조ListTargetsForPolicy의 섹션을 참조하세요. AWS CLI

### `list-targets-for-security-profile`

다음 코드 예시에서는 `list-targets-for-security-profile`을 사용하는 방법을 보여 줍니다.

AWS CLI

보안 프로파일이 적용되는 대상을 나열하려면

다음 `list-targets-for-security-profile` 예제에서는 이름이 인 AWS IoT Device Defender 보안 프로파일PossibleIssue이 적용되는 대상을 나열합니다.

```
aws iot list-targets-for-security-profile \
    --security-profile-name Testprofile
```

출력:

```
{
    "securityProfileTargets": [
        {
            "arn": "arn:aws:iot:us-west-2:123456789012:all/unregistered-things"
        },
        {
            "arn": "arn:aws:iot:us-west-2:123456789012:all/registered-things"
        }
    ]
}
```

자세한 내용은 AWS IoT 개발자 안내서의 Detect Commands를 참조하세요.

- 자세한 API 내용은 명령 참조ListTargetsForSecurityProfile의 섹션을 참조하세요. AWS CLI

### `list-thing-groups-for-thing`

다음 코드 예시에서는 `list-thing-groups-for-thing`을 사용하는 방법을 보여 줍니다.

AWS CLI

사물이 속한 그룹을 나열하려면

다음 list-thing-groups-for-thing 예제에서는 지정된 사물이 속한 그룹을 나열합니다.

```
aws iot list-thing-groups-for-thing \
    --thing-name MyLightBulb
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 IoT 개발자 안내서의 사물 그룹을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조ListThingGroupsForThing의 섹션을 참조하세요. AWS CLI

### list-thing-groups

다음 코드 예시에서는 list-thing-groups을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 계정에 정의된 사물 그룹을 나열하려면

다음 describe-thing-group 예제에서는 AWS 계정에 정의된 모든 사물 그룹을 나열합니다.

```
aws iot list-thing-groups
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 IoT 개발자 안내서의 사물 그룹을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조ListThingGroups의 섹션을 참조하세요. AWS CLI

### `list-thing-principals`

다음 코드 예시에서는 `list-thing-principals`을 사용하는 방법을 보여 줍니다.

AWS CLI

사물과 연결된 보안 주체를 나열하려면

다음 `list-thing-principals` 예제에서는 지정된 사물과 연결된 보안 주체(X.509 인증서, IAM 사용자, 그룹, 역할, Amazon Cognito 자격 증명 또는 페더레이션 자격 증명)를 나열합니다.

```
aws iot list-thing-principals \
    --thing-name MyRaspberryPi
```

출력:

```
{
    "principals": [
        "arn:aws:iot:us-
west-2:123456789012:cert/33475ac865079a5ffd5ecd44240640349293facc760642d7d8d5dbb6b4c86893"
    ]
}
```

자세한 내용은 IoT 참조ListThingPrincipals의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조ListThingPrincipals의 섹션을 참조하세요. AWS CLI

### list-thing-types

다음 코드 예시에서는 list-thing-types을 사용하는 방법을 보여 줍니다.

AWS CLI

정의된 사물 유형을 나열하려면

다음 list-thing-types 예제에서는 AWS 계정에 정의된 사물 유형의 목록을 표시합니다.

```
aws iot list-thing-types
```

출력:

```
{
    "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
            }
        }
    ]
}
```

자세한 내용은 IoT 개발자 안내서의 사물 유형을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조ListThingTypes의 섹션을 참조하세요. AWS CLI

### list-things-in-billing-group

다음 코드 예시에서는 list-things-in-billing-group을 사용하는 방법을 보여 줍니다.

AWS CLI

결제 그룹의 사물을 나열하려면

다음 `list-things-in-billing-group` 예제에서는 지정된 결제 그룹에 있는 사물을 나열합니다.

```
aws iot list-things-in-billing-group \
    --billing-group-name GroupOne
```

출력:

```
{
    "things": [
        "MyOtherLightBulb",
        "MyLightBulb"
    ]
}
```

자세한 내용은 IoT 개발자 안내서의 결제 그룹을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조ListThingsInBillingGroup의 섹션을 참조하세요. AWS CLI

### `list-things-in-thing-group`

다음 코드 예시에서는 `list-things-in-thing-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

그룹에 속하는 사물을 나열하려면

다음 `list-things-in-thing-group` 예제에서는 지정된 사물 그룹에 속하는 사물을 나열합니다.

```
aws iot list-things-in-thing-group \
    --thing-group-name LightBulbs
```

출력:

```
{
    "things": [
        "MyLightBulb"
```

```
    ]
}
```

자세한 내용은 IoT 개발자 안내서의 사물 그룹을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조ListThingsInThingGroup의 섹션을 참조하세요. AWS CLI

### `list-things`

다음 코드 예시에서는 `list-things`을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: 레지스트리의 모든 항목을 나열하는 방법

다음 `list-things` 예제에서는 AWS 계정의 AWS IoT 레지스트리에 정의된 사물(장치)을 나열합니다.

```
aws iot list-things
```

출력:

```
{
    "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
        }
    ]
}
```

예 2: 특정 속성을 가진 정의된 항목을 나열하는 방법

다음 `list-things` 예시에서는 이름이 `wattage`인 속성을 가진 사물의 목록을 표시합니다.

```
aws iot list-things \
    --attribute-name wattage
```

출력:

```
{
    "things": [
        {
            "thingName": "MyLightBulb",
            "thingTypeName": "LightBulb",
            "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyLightBulb",
            "attributes": {
                "model": "123",
                "wattage": "75"
            },
            "version": 1
        },
        {
            "thingName": "MyOtherLightBulb",
```

```
            "thingTypeName": "LightBulb",
            "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyOtherLightBulb",
            "attributes": {
                "model": "123",
                "wattage": "75"
            },
            "version": 3
        }
    ]
}
```

자세한 내용은 AWS IoT 개발자 안내서의 레지스트리를 사용하여 사물을 관리하는 방법을 참조하세요.

- 자세한 API 내용은 명령 참조ListThings의 섹션을 참조하세요. AWS CLI

### `list-topic-rule-destinations`

다음 코드 예시에서는 `list-topic-rule-destinations`을 사용하는 방법을 보여 줍니다.

AWS CLI

주제 규칙 대상을 나열하려면

다음 `list-topic-rule-destinations` 예제에서는 현재 AWS 리전에서 정의한 모든 주제 규칙 대상을 나열합니다.

```
aws iot list-topic-rule-destinations
```

출력:

```
{
    "destinationSummaries": [
        {
            "arn": "arn:aws:iot:us-west-2:123456789012:ruledestination/http/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
            "status": "ENABLED",
            "httpUrlSummary": {
                "confirmationUrl": "https://example.com"
            }
        }
    ]
```

```
}
```

자세한 내용은 AWS IoT 개발자 안내서의 주제 규칙 대상 작업을 참조하세요.

- 자세한 API 내용은 명령 참조ListTopicRuleDestinations의 섹션을 참조하세요. AWS CLI

### `list-topic-rules`

다음 코드 예시에서는 `list-topic-rules`을 사용하는 방법을 보여 줍니다.

AWS CLI

규칙을 나열하려면

다음 `list-topic-rules` 예제에서는 정의한 모든 규칙을 나열합니다.

```
aws iot list-topic-rules
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 IoT 개발자 안내서의 규칙 보기를 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조ListTopicRules의 섹션을 참조하세요. AWS CLI

### `list-v2-logging-levels`

다음 코드 예시에서는 `list-v2-logging-levels`을 사용하는 방법을 보여 줍니다.

AWS CLI

로깅 수준을 나열하려면

다음 `list-v2-logging-levels` 예제에서는 구성된 로깅 수준을 나열합니다. 로깅 수준이 설정되지 않은 경우 이 명령을 실행할 때 이 `NotConfiguredException` 발생합니다.

```
aws iot list-v2-logging-levels
```

출력:

```
{
    "logTargetConfigurations": [
        {
            "logTarget": {
                "targetType": "DEFAULT"
            },
            "logLevel": "ERROR"
        }
    ]
}
```

- API 자세한 내용은 AWS CLI 명령 참조의 ListV2LoggingLevels를 참조하세요.

### `list-violation-events`

다음 코드 예시에서는 `list-violation-events`을 사용하는 방법을 보여 줍니다.

AWS CLI

특정 기간 동안 보안 프로필 위반을 나열하려면

다음 `list-violation-events` 예제에서는 현재 AWS 계정 및 AWS 리전의 모든 AWS IoT Device Defender 보안 프로파일에 대해 2019년 6월 5일부터 2019년 6월 12일까지 발생한 위반을 나열합니다.

```
aws iot list-violation-events \
    --start-time 1559747125 \
    --end-time 1560351925
```

출력:

```
{
    "violationEvents": [
        {
            "violationId": "174db59167fa474c80a652ad1583fd44",
            "thingName": "iotconsole-1560269126751-1",
            "securityProfileName": "Testprofile",
            "behavior": {
                "name": "Authorization",
                "metric": "aws:num-authorization-failures",
                "criteria": {
                    "comparisonOperator": "greater-than",
                    "value": {
                        "count": 10
                    },
                    "durationSeconds": 300,
                    "consecutiveDatapointsToAlarm": 1,
                    "consecutiveDatapointsToClear": 1
                }
            },
            "metricValue": {
                "count": 0
            },
            "violationEventType": "in-alarm",
            "violationEventTime": 1560279000.0
        },
        {
            "violationId": "c8a9466a093d3b7b35cd44ca58bdbeab",
            "thingName": "TvnQoEoU",
            "securityProfileName": "Testprofile",
            "behavior": {
                "name": "CellularBandwidth",
                "metric": "aws:message-byte-size",
                "criteria": {
                    "comparisonOperator": "greater-than",
                    "value": {
                        "count": 128
                    },
```

```
                "consecutiveDatapointsToAlarm": 1,
                "consecutiveDatapointsToClear": 1
            }
        },
        "metricValue": {
            "count": 110
        },
        "violationEventType": "in-alarm",
        "violationEventTime": 1560276600.0
    },
    {
        "violationId": "74aa393adea02e6648f3ac362beed55e",
        "thingName": "iotconsole-1560269232412-2",
        "securityProfileName": "Testprofile",
        "behavior": {
            "name": "Authorization",
            "metric": "aws:num-authorization-failures",
            "criteria": {
                "comparisonOperator": "greater-than",
                "value": {
                    "count": 10
                },
                "durationSeconds": 300,
                "consecutiveDatapointsToAlarm": 1,
                "consecutiveDatapointsToClear": 1
            }
        },
        "metricValue": {
            "count": 0
        },
        "violationEventType": "in-alarm",
        "violationEventTime": 1560276600.0
    },
    {
        "violationId": "1e6ab5f7cf39a1466fcd154e1377e406",
        "thingName": "TvnQoEoU",
        "securityProfileName": "Testprofile",
        "behavior": {
            "name": "Authorization",
            "metric": "aws:num-authorization-failures",
            "criteria": {
                "comparisonOperator": "greater-than",
                "value": {
                    "count": 10
```

```
                    },
                    "durationSeconds": 300,
                    "consecutiveDatapointsToAlarm": 1,
                    "consecutiveDatapointsToClear": 1
                }
            },
            "metricValue": {
                "count": 0
            },
            "violationEventType": "in-alarm",
            "violationEventTime": 1560276600.0
        }
    ]
}
```

자세한 내용은 AWS IoT 개발자 안내서의 Detect Commands를 참조하세요.

- 자세한 API 내용은 명령 참조ListViolationEvents의 섹션을 참조하세요. AWS CLI

### register-ca-certificate

다음 코드 예시에서는 register-ca-certificate을 사용하는 방법을 보여 줍니다.

AWS CLI

CA(인증 기관) 인증서를 등록하려면

다음 register-ca-certificate 예제에서는 CA 인증서를 등록합니다. 명령은 CA 인증서와 CA 인증서와 연결된 프라이빗 키를 소유했음을 증명하는 키 확인 인증서를 제공합니다.

```
aws iot register-ca-certificate \
    --ca-certificate file://rootCA.pem \
    --verification-cert file://verificationCert.pem
```

출력:

```
{
    "certificateArn": "arn:aws:iot:us-west-2:123456789012:cacert/
f4efed62c0142f16af278166f61962501165c4f053629520742646005 8cd1467",
    "certificateId":
 "f4efed62c0142f16af278166f61962501165c4f053629520742646005 8cd1467"
 }
```

자세한 내용은 AWS IoT API 참조의 RegisterCACertificate을 참조하세요.

- 자세한 API 내용은 명령 참조RegisterCaCertificate의 섹션을 참조하세요. AWS CLI

### register-certificate

다음 코드 예시에서는 register-certificate을 사용하는 방법을 보여 줍니다.

AWS CLI

자체 서명된 디바이스 인증서를 등록하려면

다음 register-certificate 예제에서는 rootCA.pem CA 인증서로 서명된 deviceCert.pem 디바이스 인증서를 등록합니다. CA 인증서를 사용하여 자체 서명된 디바이스 인증서를 등록하려면 먼저 CA 인증서를 등록해야 합니다. 자체 서명된 인증서는 이 명령에 전달하는 것과 동일한 CA 인증서로 서명해야 합니다.

```
aws iot register-certificate \
    --certificate-pem file://deviceCert.pem \
    --ca-certificate-pem file://rootCA.pem
```

출력:

```
{
    "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142",
    "certificateId":
 "488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142"
 }
```

자세한 내용은 IoT 참조RegisterCertificate의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조RegisterCertificate의 섹션을 참조하세요. AWS CLI

### register-thing

다음 코드 예시에서는 register-thing을 사용하는 방법을 보여 줍니다.

AWS CLI

사물을 등록하려면

다음 register-thing 예제에서는 프로비저닝 템플릿을 사용하여 사물을 등록합니다.

```
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
```

출력:

```
{
    "certificatePem": "-----BEGIN CERTIFICATE-----
\nMIIDWTCCAkGgAwIBAgIUYLk81I35cIppobpw
HiOJ2jNjboIwDQYJKoZIhvcNAQEL
\nBQAwTTFLMEkGA1UECwxCQW1hem9uIFdlYiBTZXJ2aWNlcyBPPUFtYXpvbi
5jb20g\nSW5jLiBMPVNlYXR0bGUgU1Q9V2FzaGluZ3RvbiBDPVVTMB4XDTIwMDcyMzE2NDUw
\nOVoXDTQ5MTIzMT
IzNTk1OVowHjEcMBoGA1UEAwwTQVdTIElvVCBDZXJ0aWZpY2F0\nZTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCC
AQoCggEBAO71uADhdBajqTmgrMV5\nmCFfBZQRMo1MdtVoZr2X+M4MzL
+RARrtUzH9a2SMAckeX8KeblIOTKzORI
RDXnyE
\n6lVOwjgAsd0ku22rFxex4eG2ikha7pYYkvuToqA7L3TxItRvfKrxRI4ZfJoFPip4\nKqiuBJVNOGKTcQ
Hd1RNOrddwwu6kFJLeKDmEXAMPLEdUF0N+qfR9yKnZQkm
+g6Q2\nGXu7u0W3hn6nlRN8qVoka0uW12p53xM7oHVz
Gf+cxKBxlbOhGkp6yCfTSkUBm3Sp\n9zLw35kiHXVm4EVpwgNlnk6XcIGIkw8a/iy4pzmvuGAANY1/uU/
zgCjymw
ZT5S30\nBV0CAwEAAaNgMF4wHwYDVR0jBBgwFoAUGx0tCcU3q2n1WXAuUCv6hugXjKswHQYD
\nVR0OBBYEFOVtvZ
9Aj2RYFnkX7Iu01XTRUdxgMAwGA1UdEwEB/wQCMAAwDgYDVR0P\nAQH/
BAQDAgeAMA0GCSqGSIb3DQEBCwUAA4IB
AQCXCQcpOtubS5ftOsDMTcpP/jNX
\nDHyArxmjpSc2aCdmm7WX59lTKWyAdxGAvqaDVWqTo0oXI7tZ8w7aINlGi5
pXnifx\n3SBebMUoBbTktrC97yUaeL025mCFv8emDnTR/fE7PTsBKjW0g/rrfpwBxZLXDFwN
\nnqkQjy3EDfifj2
6j0xYIqqWMPogyn4srOCKynS5wMJuQZlHQOnabVwnwK4Y0Mflp
\np9+4susFUR9aT3BT1AcIwqSpzhlKhh4Iz7ND
```

```
kRn4amsUT210jg/zOO1Ow+BTHcVQ\nJly8XDu0CWSu04q6SnaBzHmlySIajxuRTP/AdfRouP1OXe
+qlbPOBcvVvF
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"
    }
}
```

자세한 내용은 IoT Core 개발자 안내서의 신뢰할 수 있는 사용자별 프로비저닝을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조RegisterThing의 섹션을 참조하세요. AWS CLI

### `reject-certificate-transfer`

다음 코드 예시에서는 `reject-certificate-transfer`을 사용하는 방법을 보여 줍니다.

AWS CLI

인증서 전송을 거부하려면

다음 `reject-certificate-transfer` 예제에서는 다른 AWS 계정에서 지정된 디바이스 인증서의 전송을 거부합니다.

```
aws iot reject-certificate-transfer \
    --certificate-
id f0f33678c7c9a046e5cc87b2b1a58dfa0beec26db78addd5e605d630e05c7fc8
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT Core 개발자 안내서의 다른 계정으로 인증서 전송을 참조하세요.

- 자세한 API 내용은 명령 참조RejectCertificateTransfer의 섹션을 참조하세요. AWS CLI

### `remove-thing-from-billing-group`

다음 코드 예시에서는 `remove-thing-from-billing-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

결제 그룹에서 사물을 제거하려면

다음 remove-thing-from-billing-group 예제에서는 결제 그룹에서 지정된 사물을 제거합니다.

```
aws iot remove-thing-from-billing-group \
    --billing-group-name GroupOne \
    --thing-name MyOtherLightBulb
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IoT 개발자 안내서의 결제 그룹을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조RemoveThingFromBillingGroup의 섹션을 참조하세요. AWS CLI

### remove-thing-from-thing-group

다음 코드 예시에서는 remove-thing-from-thing-group을 사용하는 방법을 보여 줍니다.

AWS CLI

사물 그룹에서 사물을 제거하려면

다음 remove-thing-from-thing-group 예제에서는 사물 그룹에서 지정된 사물을 제거합니다.

```
aws iot remove-thing-from-thing-group \
    --thing-name bulb7 \
    --thing-group-name DeadBulbs
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 사물 그룹 <https://docs.aws.amazon.com/iot/latest/developerguide/thing-groups.html >을 참조하세요.

- 자세한 API 내용은 명령 참조RemoveThingFromThingGroup의 섹션을 참조하세요. AWS CLI

### replace-topic-rule

다음 코드 예시에서는 replace-topic-rule을 사용하는 방법을 보여 줍니다.

AWS CLI

주제의 규칙 정의를 업데이트하려면

다음 replace-topic-rule 예제에서는 지정된 규칙을 업데이트하여 토양 수분 수준 판독값이 너무 낮을 때 SNS 알림을 보냅니다.

```
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\"}"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 IoT 규칙 생성을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조ReplaceTopicRule의 섹션을 참조하세요. AWS CLI

### search-index

다음 코드 예시에서는 search-index을 사용하는 방법을 보여 줍니다.

AWS CLI

사물 인덱스를 쿼리하려면

다음 search-index 예제에서는 의 유형이 있는 사물에 대해 AWS_Things 인덱스를 쿼리합니다LightBulb.

```
aws iot search-index \
    --index-name "AWS_Things" \
    --query-string "thingTypeName:LightBulb"
```

출력:

```
{
    "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
            }
        }
    ]
}
```

자세한 내용은 AWS IoT 개발자 안내서의 사물 인덱싱 관리를 참조하세요.

- 자세한 API 내용은 명령 참조SearchIndex의 섹션을 참조하세요. AWS CLI

### set-default-authorizer

다음 코드 예시에서는 set-default-authorizer을 사용하는 방법을 보여 줍니다.

AWS CLI

기본 권한 부여자를 설정하려면

다음 set-default-authorizer 예제에서는 CustomAuthorizer라는 사용자 지정 권한 부여자를 기본 권한 부여자로 설정합니다.

```
aws iot set-default-authorizer \
    --authorizer-name CustomAuthorizer
```

출력:

```
{
    "authorizerName": "CustomAuthorizer",
    "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/
CustomAuthorizer"
}
```

자세한 내용은 IoT 참조CreateDefaultAuthorizer의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조SetDefaultAuthorizer의 섹션을 참조하세요. AWS CLI

### set-default-policy-version

다음 코드 예시에서는 set-default-policy-version을 사용하는 방법을 보여 줍니다.

AWS CLI

정책의 기본 버전을 설정하려면

다음 set-default-policy-version 예제에서는 라는 정책의 기본 버전을 2로 설정합니다UpdateDeviceCertPolicy.

```
aws iot set-default-policy-version \
    --policy-name UpdateDeviceCertPolicy \
    --policy-version-id 2
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조SetDefaultPolicyVersion의 섹션을 참조하세요. AWS CLI

### set-v2-logging-level

다음 코드 예시에서는 set-v2-logging-level을 사용하는 방법을 보여 줍니다.

AWS CLI

사물 그룹에 대한 로깅 수준을 설정하려면

다음 set-v2-logging-level 예제에서는 로깅 수준을 지정된 사물 그룹에 대한 경고를 로그하도록 설정합니다.

```
aws iot set-v2-logging-level \
    --log-target "{\"targetType\":\"THING_GROUP\",\"targetName\":\"LightBulbs\"}" \
    --log-level WARN
```

이 명령은 출력을 생성하지 않습니다.

- API 자세한 내용은 AWS CLI 명령 참조의 SetV2LoggingLevel를 참조하세요.

### set-v2-logging-options

다음 코드 예시에서는 set-v2-logging-options을 사용하는 방법을 보여 줍니다.

AWS CLI

로깅 옵션을 설정하려면

다음 set-v2-logging-options 예제에서는 기본 로깅 세부 정보 수준을 ERROR로 설정하고 로깅에 사용할 ARN를 지정합니다.

```
aws iot set-v2-logging-options \
    --default-log-level ERROR \
    --role-arn "arn:aws:iam::094249569039:role/service-role/iotLoggingRole"
```

이 명령은 출력을 생성하지 않습니다.

- API 자세한 내용은 AWS CLI 명령 참조의 SetV2LoggingOptions를 참조하세요.

### start-audit-mitigation-actions-task

다음 코드 예시에서는 start-audit-mitigation-actions-task을 사용하는 방법을 보여 줍니다.

AWS CLI

감사 결과에 완화 조치를 적용하려면

다음 start-audit-mitigation-actions-task 예제에서는 지정된 단일 결과에 ResetPolicyVersionAction 작업(정책을 삭제함)을 적용합니다.

```
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"
```

출력:

```
{
    "taskId": "myActionsTaskId"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 StartAuditMitigationActionsTask (완화 작업 명령)을 참조하세요.

- 자세한 API 내용은 명령 참조StartAuditMitigationActionsTask의 섹션을 참조하세요. AWS CLI

### start-on-demand-audit-task

다음 코드 예시에서는 start-on-demand-audit-task을 사용하는 방법을 보여 줍니다.

AWS CLI

즉시 감사를 시작하려면

다음 start-on-demand-audit-task 예제에서는 AWS IoT Device Defender 감사를 시작하고 세 가지 인증서 검사를 수행합니다.

```
aws iot start-on-demand-audit-task \
```

```
    --target-check-
names CA_CERTIFICATE_EXPIRING_CHECK DEVICE_CERTIFICATE_EXPIRING_CHECK REVOKED_CA_CERTIFICATE
```

출력:

```
{
    "taskId": "a3aea009955e501a31b764abe1bebd3d"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 감사 명령을 참조하세요.

- 자세한 API 내용은 명령 참조StartOnDemandAuditTask의 섹션을 참조하세요. AWS CLI

### tag-resource

다음 코드 예시에서는 tag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스의 태그 키 및 값을 지정하려면

다음 tag-resource 예제에서는 키를 사용하여 태그를 적용Assembly하고 사물 그룹 Fact1NW에 값을 적용합니다LightBulbs.

```
aws iot tag-resource \
    --tags Key=Assembly,Value="Fact1NW" \
    --resource-arn "arn:aws:iot:us-west-2:094249569039:thinggroup/LightBulbs"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 IoT 리소스 태그 지정을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### test-authorization

다음 코드 예시에서는 test-authorization을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS IoT 정책을 테스트하려면

다음 test-authorization 예제에서는 지정된 보안 주체와 연결된 AWS IoT 정책을 테스트합니다.

```
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
```

출력:

```
{
    "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": []
        }
    ]
}
```

자세한 내용은 IoT 참조TestAuthorization의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조TestAuthorization의 섹션을 참조하세요. AWS CLI

### `test-invoke-authorizer`

다음 코드 예시에서는 `test-invoke-authorizer`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 권한 부여자를 테스트하려면

다음 `test-invoke-authorizer` 예제는 사용자 지정 권한 부여자를 testS.

```
aws iot test-invoke-authorizer \
    --authorizer-name IoTAuthorizer \
    --token allow \
    --token-signature "mE0GvaHqy9nER/
FdgtJX5lXYEJ3b3vE7t1gEszc0TKGgLKWXTnPkb2AbKnOAZ8lGyoN5dVtWDWVmr25m7+
+zjbYIMk2TBvyGXhOmvKFBPkdgyA43KL6SiZy0cTqlPMcQDsP7VX2rXr7CTowCxSNKphGXdQe0/
I5dQ+JO6KUaHwCmupt0/MejKtaNwiia064j6wprOAUwG5S1IYFuRd0X
+wfo8pb0DubAIX1Ua705kuhRUcTx4SxUShEYKmN4IDEvLB6FsIr0B2wvB7y4iPmcajxzGl02ExvyCUNctCV9dYlRRGJj
```

출력:

```
{
    "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
}
```

자세한 내용은 IoT 참조TestInvokeAuthorizer의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조TestInvokeAuthorizer의 섹션을 참조하세요. AWS CLI

### `transfer-certificate`

다음 코드 예시에서는 `transfer-certificate`을 사용하는 방법을 보여 줍니다.

AWS CLI

디바이스 인증서를 다른 AWS 계정으로 전송하려면

다음 `transfer-certificate` 예제에서는 디바이스 인증서를 다른 AWS 계정으로 전송합니다. 인증서와 AWS 계정은 ID로 식별됩니다.

```
aws iot transfer-certificate \
    --certificate-
id 488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142 \
    --target-aws-account 030714055129
```

출력:

```
{
    "transferredCertificateArn": "arn:aws:iot:us-
west-2:030714055129:cert/488b6a7f2acdeb00a77384e63c4e40b18b1b3caaae57b7272ba44c45e3448142"
}
```

자세한 내용은 AWS IoT Core 개발자 안내서의 다른 계정으로 인증서 전송을 참조하세요.

- 자세한 API 내용은 명령 참조TransferCertificate의 섹션을 참조하세요. AWS CLI

### `untag-resource`

다음 코드 예시에서는 `untag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에서 태그 키를 제거하려면

다음 `untag-resource` 예제에서는 사물 그룹 에서 태그`MyTag`와 해당 값을 제거합니다`LightBulbs`.

```
command
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 IoT 리소스 태그 지정을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-account-audit-configuration

다음 코드 예시에서는 `update-account-audit-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 감사 SNS 알림에 Amazon 알림을 활성화하려면

다음 `update-account-audit-configuration` 예제에서는 AWS IoT Device Defender 감사 SNS 알림에 대한 Amazon 알림을 활성화하여 대상과 해당 대상에 쓰는 데 사용되는 역할을 지정합니다.

```
aws iot update-account-audit-configuration \
    --audit-notification-target-configurations "SNS={targetArn=\"arn:aws:sns:us-west-2:123456789012:ddaudits\",roleArn=\"arn:aws:iam::123456789012:role/service-role/AWSIoTDeviceDefenderAudit\",enabled=true}"
```

이 명령은 출력을 생성하지 않습니다.

예제 2: 감사 확인을 활성화하려면

다음 `update-account-audit-configuration` 예제에서는 라는 AWS IoT Device Defender 감사 확인을 활성화합니다`AUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK`. AWS 계정에 대해 하나 이상의 예약된 감사에 `targetCheckNames` 대해 의 일부인 경우 감사 확인을 비활성화할 수 없습니다.

```
aws iot update-account-audit-configuration \
    --audit-check-configurations
 "{\"AUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK\":{\"enabled\":true}}"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 감사 명령을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateAccountAuditConfiguration의 섹션을 참조하세요. AWS CLI

### update-audit-suppression

다음 코드 예시에서는 update-audit-suppression을 사용하는 방법을 보여 줍니다.

AWS CLI

감사 결과 금지를 업데이트하려면

다음 update-audit-suppression 예제에서는 감사 결과 금지의 만료 날짜를 2020-09-21로 업데이트합니다.

```
aws iot update-audit-suppression \
    --check-name DEVICE_CERTIFICATE_EXPIRING_CHECK \
    --resource-identifier deviceCertificateId=c7691e<shortened> \
    --no-suppress-indefinitely \
    --expiration-date 2020-09-21
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IoT 개발자 안내서의 감사 결과 금지를 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조UpdateAuditSuppression의 섹션을 참조하세요. AWS CLI

### update-authorizer

다음 코드 예시에서는 update-authorizer을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 권한 부여자를 업데이트하려면

다음 update-authorizer 예제는 에서 CustomAuthorizer2의 상태입니다INACTIVE.

```
aws iot update-authorizer \
    --authorizer-name CustomAuthorizer2 \
    --status INACTIVE
```

출력:

```
{
    "authorizerName": "CustomAuthorizer2",
    "authorizerArn": "arn:aws:iot:us-west-2:123456789012:authorizer/
CustomAuthorizer2"
}
```

자세한 내용은 IoT 참조UpdateAuthorizer의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조UpdateAuthorizer의 섹션을 참조하세요. AWS CLI

### update-billing-group

다음 코드 예시에서는 update-billing-group을 사용하는 방법을 보여 줍니다.

AWS CLI

결제 그룹에 대한 정보를 업데이트하려면

다음 update-billing-group 예제에서는 지정된 결제 그룹에 대한 설명을 업데이트합니다.

```
aws iot update-billing-group \
    --billing-group-name GroupOne \
    --billing-group-properties "billingGroupDescription=\"Primary bulb billing group
\""
```

출력:

```
{
    "version": 2
}
```

자세한 내용은 IoT 개발자 안내서의 결제 그룹을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조UpdateBillingGroup의 섹션을 참조하세요. AWS CLI

### update-ca-certificate

다음 코드 예시에서는 update-ca-certificate을 사용하는 방법을 보여 줍니다.

AWS CLI

CA(인증 기관) 인증서를 업데이트하려면

다음 update-ca-certificate 예제에서는 지정된 CA 인증서를 ACTIVE 상태로 설정합니다.

```
aws iot update-ca-certificate \
    --certificate-
id f4efed62c0142f16af278166f61962501165c4f0536295207426460058cd1467 \
    --new-status ACTIVE
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT API 참조의 UpdateCACertificate를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateCaCertificate의 섹션을 참조하세요. AWS CLI

### update-certificate

다음 코드 예시에서는 update-certificate을 사용하는 방법을 보여 줍니다.

AWS CLI

디바이스 인증서를 업데이트하려면

다음 update-certificate 예제에서는 지정된 디바이스 인증서를 INACTIVE 상태로 설정합니다.

```
aws iot update-certificate \
    --certificate-
id d1eb269fb55a628552143c8f96eb3c258fcd5331ea113e766ba0c82bf225f0be \
    --new-status INACTIVE
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IoT 참조UpdateCertificate의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조UpdateCertificate의 섹션을 참조하세요. AWS CLI

### update-custom-metric

다음 코드 예시에서는 update-custom-metric을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 지표를 업데이트하려면

다음 update-custom-metric 예제에서는 사용자 지정 지표를 새 로 업데이트합니다display-name.

```
aws iot update-custom-metric \
    --metric-name batteryPercentage \
    --display-name 'remaining battery percentage on device' \
    --region us-east-1
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS IoT Core 개발자 안내서의 사용자 지정 지표를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateCustomMetric의 섹션을 참조하세요. AWS CLI

### update-dimension

다음 코드 예시에서는 update-dimension을 사용하는 방법을 보여 줍니다.

AWS CLI

차원을 업데이트하려면

다음 update-dimension 예제에서는 차원을 업데이트합니다.

```
aws iot update-dimension \
```

```
    --name TopicFilterForAuthMessages \
    --string-values device/${iot:ClientId}/auth
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS IoT Core 개발자 안내서의 차원을 사용하여 보안 프로필의 지표 범위 지정을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateDimension의 섹션을 참조하세요. AWS CLI

### update-domain-configuration

다음 코드 예시에서는 update-domain-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

도메인 구성을 업데이트하려면

다음 update-domain-configuration 예제에서는 지정된 도메인 구성을 비활성화합니다.

```
aws iot update-domain-configuration \
    --domain-configuration-name "additionalDataDomain" \
    --domain-configuration-status "DISABLED"
```

출력:

```
{
    "domainConfigurationName": "additionalDataDomain",
    "domainConfigurationArn": "arn:aws:iot:us-
west-2:123456789012:domainconfiguration/additionalDataDomain/dikMh"
```

```
}
```

자세한 내용은 AWS IoT 개발자 안내서의 구성 가능한 엔드포인트를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateDomainConfiguration의 섹션을 참조하세요. AWS CLI

### update-dynamic-thing-group

다음 코드 예시에서는 update-dynamic-thing-group을 사용하는 방법을 보여 줍니다.

AWS CLI

동적 사물 그룹을 업데이트하려면

다음 update-dynamic-thing-group 예제에서는 지정된 동적 사물 그룹을 업데이트합니다. 설명을 제공하고 쿼리 문자열을 업데이트하여 그룹 멤버십 기준을 변경합니다.

```
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"
```

출력:

```
{
    "version": 2
}
```

자세한 내용은 IoT 개발자 안내서의 동적 사물 그룹을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조UpdateDynamicThingGroup의 섹션을 참조하세요. AWS CLI

### update-event-configurations

다음 코드 예시에서는 update-event-configurations을 사용하는 방법을 보여 줍니다.

AWS CLI

게시되는 이벤트 유형을 표시하려면

다음 update-event-configurations 예제에서는 CA 인증서가 추가, 업데이트 또는 삭제될 때 메시지를 활성화하도록 구성을 업데이트합니다.

```
aws iot update-event-configurations \
    --event-configurations "{\"CA_CERTIFICATE\":{\"Enabled\":true}}"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 이벤트 메시지를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateEventConfigurations의 섹션을 참조하세요. AWS CLI

### update-indexing-configuration

다음 코드 예시에서는 update-indexing-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

사물 인덱싱을 활성화하려면

다음 update-indexing-configuration 예제에서는 사물 인덱싱이 AWS_Things 인덱스를 사용하여 레지스트리 데이터, 섀도우 데이터 및 사물 연결 상태 검색을 지원할 수 있습니다.

```
aws iot update-indexing-configuration
    --thing-indexing-
configuration thingIndexingMode=REGISTRY_AND_SHADOW,thingConnectivityIndexingMode=STATUS
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IoT 개발자 안내서의 사물 인덱싱 관리를 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조UpdateIndexingConfiguration의 섹션을 참조하세요. AWS CLI

### update-job

다음 코드 예시에서는 update-job을 사용하는 방법을 보여 줍니다.

AWS CLI

작업에 대한 세부 상태를 가져오려면

다음 update-job 예제에서는 ID가 인 작업에 대한 세부 상태를 가져옵니다example-job-01.

```
aws iot describe-job \
```

```
    --job-id "example-job-01"
```

출력:

```
{
    "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": {}
    }
}
```

자세한 내용은 AWS IoT 개발자 안내서의 작업 생성 및 관리(CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateJob의 섹션을 참조하세요. AWS CLI

### update-mitigation-action

다음 코드 예시에서는 update-mitigation-action을 사용하는 방법을 보여 줍니다.

AWS CLI

완화 조치를 업데이트하려면

다음 update-mitigation-action 예제에서는 라는 지정된 완화 작업을 업데이트하고 사물 그룹 이름을 AddThingsToQuarantineGroupAction변경한 다음 를 overrideDynamicGroups로 설정합니다false. describe-mitigation-action 명령을 사용하여 변경 사항을 확인할 수 있습니다.

```
aws iot update-mitigation-action \
    --cli-input-json "{ \"actionName\": \"AddThingsToQuarantineGroupAction\",
 \"actionParams\": { \"addThingsToThingGroupParams\": {\"thingGroupNames\":
[\"QuarantineGroup2\"],\"overrideDynamicGroups\": false}}}"
```

출력:

```
{
    "actionArn": "arn:aws:iot:us-west-2:123456789012:mitigationaction/
AddThingsToQuarantineGroupAction",
    "actionId": "2fd2726d-98e1-4abf-b10f-09465ccd6bfa"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 UpdateMitigationAction (완화 작업 명령)을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateMitigationAction의 섹션을 참조하세요. AWS CLI

### update-provisioning-template

다음 코드 예시에서는 update-provisioning-template을 사용하는 방법을 보여 줍니다.

AWS CLI

프로비저닝 템플릿을 업데이트하려면

다음 update-provisioning-template 예제에서는 지정된 프로비저닝 템플릿에 대한 설명과 역할 Arn을 수정하고 템플릿을 활성화합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT Core 개발자 안내서의 IoT Secure Tunneling을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조UpdateProvisioningTemplate의 섹션을 참조하세요. AWS CLI

### update-role-alias

다음 코드 예시에서는 update-role-alias을 사용하는 방법을 보여 줍니다.

AWS CLI

역할 별칭을 업데이트하려면

다음 update-role-alias 예제에서는 LightBulbRole 역할 별칭을 업데이트합니다.

```
aws iot update-role-alias \
    --role-alias LightBulbRole \
    --role-arn arn:aws:iam::123456789012:role/lightbulbrole-001
```

출력:

```
{
    "roleAlias": "LightBulbRole",
    "roleAliasArn": "arn:aws:iot:us-west-2:123456789012:rolealias/LightBulbRole"
}
```

자세한 내용은 IoT 참조UpdateRoleAlias의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조UpdateRoleAlias의 섹션을 참조하세요. AWS CLI

### update-scheduled-audit

다음 코드 예시에서는 update-scheduled-audit을 사용하는 방법을 보여 줍니다.

AWS CLI

예약된 감사 정의를 업데이트하려면

다음 update-scheduled-audit 예제에서는 AWS IoT Device Defender 예약 감사의 대상 검사 이름을 변경합니다.

```
aws iot update-scheduled-audit \
    --scheduled-audit-name WednesdayCertCheck \
    --target-check-
names CA_CERTIFICATE_EXPIRING_CHECK DEVICE_CERTIFICATE_EXPIRING_CHECK REVOKED_CA_CERTIFICATE
```

출력:

```
{
    "scheduledAuditArn": "arn:aws:iot:us-west-2:123456789012:scheduledaudit/
WednesdayCertCheck"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 감사 명령을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateScheduledAudit의 섹션을 참조하세요. AWS CLI

### update-security-profile

다음 코드 예시에서는 update-security-profile을 사용하는 방법을 보여 줍니다.

AWS CLI

보안 프로필을 변경하려면

다음 update-security-profile 예제에서는 AWS IoT Device Defender 보안 프로파일에 대한 설명과 동작을 모두 업데이트합니다.

```
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}}]"
```

출력:

```
{
    "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
}
```

자세한 내용은 AWS IoT 개발자 안내서의 Detect Commands를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateSecurityProfile의 섹션을 참조하세요. AWS CLI

### `update-stream`

다음 코드 예시에서는 `update-stream`을 사용하는 방법을 보여 줍니다.

AWS CLI

스트림을 업데이트하려면

다음 `update-stream` 예제에서는 기존 스트림을 업데이트합니다. 스트림 버전은 1씩 증가합니다.

```
aws iot update-stream \
    --cli-input-json file://update-stream.json
```

update-stream.json의 콘텐츠:

```
{
    "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"
}
```

출력:

```
{
    "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
}
```

자세한 내용은 IoT 참조UpdateStream의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조UpdateStream의 섹션을 참조하세요. AWS CLI

### update-thing-group

다음 코드 예시에서는 update-thing-group을 사용하는 방법을 보여 줍니다.

AWS CLI

사물 그룹에 대한 정의를 업데이트하려면

다음 update-thing-group 예제에서는 지정된 사물 그룹에 대한 정의를 업데이트하여 설명과 두 속성을 변경합니다.

```
aws iot update-thing-group \
    --thing-group-name HalogenBulbs \
    --thing-group-properties "thingGroupDescription=\"Halogen bulb group\",
 attributePayload={attributes={Manufacturer=AnyCompany,wattage=60}}"
```

출력:

```
{
    "version": 2
}
```

자세한 내용은 IoT 개발자 안내서의 사물 그룹을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조UpdateThingGroup의 섹션을 참조하세요. AWS CLI

### update-thing-groups-for-thing

다음 코드 예시에서는 update-thing-groups-for-thing을 사용하는 방법을 보여 줍니다.

AWS CLI

사물이 속한 그룹을 변경하려면

다음 update-thing-groups-for-thing 예제에서는 라는 MyLightBulb 이름의 그룹에서 라는 이름을 가진 사물을 제거하고 replaceableItems 동시에 라는 이름의 그룹에 DeadBulbs 추가합니다.

```
aws iot update-thing-groups-for-thing \
    --thing-name MyLightBulb \
    --thing-groups-to-add "replaceableItems" \
    --thing-groups-to-remove "DeadBulbs"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 사물 그룹을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateThingGroupsForThing의 섹션을 참조하세요. AWS CLI

### update-thing

다음 코드 예시에서는 update-thing을 사용하는 방법을 보여 줍니다.

AWS CLI

사물을 사물 유형과 연결하려면

다음 update-thing 예제에서는 AWS IoT 레지스트리의 사물을 사물 유형과 연결합니다. 연결하면 사물 유형으로 정의된 속성에 대한 값을 제공합니다.

```
aws iot update-thing \
    --thing-name "MyOtherLightBulb" \
    --thing-type-name "LightBulb" \
    --attribute-payload "{"attributes": {"wattage":"75", "model":"123"}}"
```

이 명령은 출력을 생성하지 않습니다. describe-thing 명령을 사용하여 결과를 확인합니다.

자세한 내용은 IoT 개발자 안내서의 사물 유형을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조UpdateThing의 섹션을 참조하세요. AWS CLI

### update-topic-rule-destination

다음 코드 예시에서는 update-topic-rule-destination을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 주제 규칙 대상 활성화

다음 update-topic-rule-destination 예제에서는 주제 규칙 대상으로의 트래픽을 활성화합니다.

```
aws iot update-topic-rule-destination \
    --arn "arn:aws:iot:us-west-2:123456789012:ruledestination/http/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE" \
    --status ENABLED
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 주제 규칙 대상 활성화를 참조하세요.

예제 2: 주제 규칙 대상을 비활성화하려면

다음 update-topic-rule-destination 예제에서는 주제 규칙 대상으로의 트래픽을 비활성화합니다.

```
aws iot update-topic-rule-destination \
    --arn "arn:aws:iot:us-west-2:123456789012:ruledestination/http/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE" \
    --status DISABLED
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 주제 규칙 대상 비활성화를 참조하세요.

예제 3: 새 확인 메시지를 보내려면

다음 update-topic-rule-destination 예제에서는 주제 규칙 대상에 대한 새 확인 메시지를 보냅니다.

```
aws iot update-topic-rule-destination \
    --arn "arn:aws:iot:us-west-2:123456789012:ruledestination/http/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE" \
    --status IN_PROGRESS
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 새 확인 메시지 전송을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateTopicRuleDestination의 섹션을 참조하세요. AWS CLI

### validate-security-profile-behaviors

다음 코드 예시에서는 validate-security-profile-behaviors을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 보안 프로필에 대한 동작 파라미터를 검증하려면

다음 validate-security-profile-behaviors 예제에서는 AWS IoT Device Defender 보안 프로파일에 대해 잘 구성되고 올바른 동작 세트를 검증합니다.

```
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}}]"
```

출력:

```
{
    "valid": true,
    "validationErrors": []
}
```

예제 2: 보안 프로필에 대해 잘못된 동작 파라미터를 검증하려면

다음 `validate-security-profile-behaviors` 예제에서는 AWS IoT Device Defender 보안 프로파일에 대한 오류가 포함된 동작 세트를 검증합니다.

```
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}}]"
```

출력:

```
{
    "valid": false,
    "validationErrors": [
        {
            "errorMessage": "Behavior Authorization is malformed.
 consecutiveDatapointsToAlarm 100000 should be in range[1,10]"
        }
    ]
}
```

자세한 내용은 AWS IoT 개발자 안내서의 Detect Commands를 참조하세요.

- 자세한 API 내용은 명령 참조ValidateSecurityProfileBehaviors의 섹션을 참조하세요. AWS CLI

## AWS IoT 1-Click 를 사용한 디바이스 예제 AWS CLI

다음 코드 예제에서는 AWS IoT 1-Click 디바이스와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `claim-devices-by-claim-code`

다음 코드 예시에서는 `claim-devices-by-claim-code`을 사용하는 방법을 보여 줍니다.

AWS CLI

클레임 코드를 사용하여 하나 이상의 AWS IoT 1-Click 디바이스를 클레임하려면

다음 `claim-devices-by-claim-code` 예제에서는 클레임 코드(디바이스 ID 대신)를 사용하여 지정된 AWS IoT 1-Click 디바이스를 클레임합니다.

```
aws iot1click-devices claim-devices-by-claim-code \
    --claim-code C-123EXAMPLE
```

출력:

```
{
   "Total": 9
   "ClaimCode": "C-123EXAMPLE"
```

```
}
```

자세한 내용은 AWS IoT 1-Click 개발자 안내서 AWS CLIAWS 의 에서 IoT 1-Click 사용을 참조하세요.

- 자세한 API 내용은 명령 참조ClaimDevicesByClaimCode의 섹션을 참조하세요. AWS CLI

### `describe-device`

다음 코드 예시에서는 `describe-device`을 사용하는 방법을 보여 줍니다.

AWS CLI

디바이스를 설명하려면

다음 `describe-device` 예제에서는 지정된 디바이스를 설명합니다.

```
aws iot1click-devices describe-device \
    --device-id G030PM0123456789
```

출력:

```
{
    "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": {}
    }
}
```

자세한 내용은 AWS IoT 1-Click 개발자 안내서 AWS CLIAWS 의 에서 IoT 1-Click 사용을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeDevice의 섹션을 참조하세요. AWS CLI

### `finalize-device-claim`

다음 코드 예시에서는 `finalize-device-claim`을 사용하는 방법을 보여 줍니다.

AWS CLI

디바이스 ID를 사용하여 AWS IoT 1-Click 디바이스에 대한 클레임 요청을 완료하려면

다음 `finalize-device-claim` 예제에서는 디바이스 ID(클레임 코드 대신)를 사용하여 지정된 AWS IoT 1-Click 디바이스에 대한 클레임 요청을 완료합니다.

```
aws iot1click-devices finalize-device-claim \
    --device-id G030PM0123456789
```

출력:

```
{
    "State": "CLAIMED"
}
```

자세한 내용은 AWS IoT 1-Click 개발자 안내서 AWS CLIAWS 의 에서 IoT 1-Click 사용을 참조하세요.

- 자세한 API 내용은 명령 참조FinalizeDeviceClaim의 섹션을 참조하세요. AWS CLI

### `get-device-methods`

다음 코드 예시에서는 `get-device-methods`을 사용하는 방법을 보여 줍니다.

AWS CLI

디바이스에 사용 가능한 메서드를 나열하려면

다음 `get-device-methods` 예제에서는 디바이스에 사용할 수 있는 방법을 나열합니다.

```
aws iot1click-devices get-device-methods \
    --device-id G030PM0123456789
```

출력:

```
{
    "DeviceMethods": [
        {
            "MethodName": "getDeviceHealthParameters"
        },
        {
            "MethodName": "setDeviceHealthMonitorCallback"
        },
        {
            "MethodName": "getDeviceHealthMonitorCallback"
        },
        {
            "MethodName": "setOnClickCallback"
        },
        {
            "MethodName": "getOnClickCallback"
        }
    ]
}
```

자세한 내용은 AWS IoT 1-Click 개발자 안내서 AWS CLIAWS 의 에서 IoT 1-Click 사용을 참조하세요.

- 자세한 API 내용은 명령 참조GetDeviceMethods의 섹션을 참조하세요. AWS CLI

### `initiate-device-claim`

다음 코드 예시에서는 `initiate-device-claim`을 사용하는 방법을 보여 줍니다.

AWS CLI

디바이스 ID를 사용하여 AWS IoT 1-Click 디바이스에 대한 클레임 요청을 시작하려면

다음 `initiate-device-claim` 예제에서는 디바이스 ID(클레임 코드 대신)를 사용하여 지정된 AWS IoT 1-Click 디바이스에 대한 클레임 요청을 시작합니다.

```
aws iot1click-devices initiate-device-claim \
    --device-id G030PM0123456789
```

출력:

```
{
```

```
    "State": "CLAIM_INITIATED"
}
```

자세한 내용은 AWS IoT 1-Click 개발자 안내서 AWS CLIAWS 의 에서 IoT 1-Click 사용을 참조하세요.

- 자세한 API 내용은 명령 참조InitiateDeviceClaim의 섹션을 참조하세요. AWS CLI

### `invoke-device-method`

다음 코드 예시에서는 `invoke-device-method`을 사용하는 방법을 보여 줍니다.

AWS CLI

디바이스에서 디바이스 메서드를 호출하려면

다음 `invoke-device-method` 예제는 디바이스에서 지정된 메서드를 호출합니다.

```
aws iot1click-devices invoke-device-method \
    --cli-input-json file://invoke-device-method.json
```

`invoke-device-method.json`의 콘텐츠:

```
{
    "DeviceId": "G030PM0123456789",
    "DeviceMethod": {
        "DeviceType": "device",
        "MethodName": "getDeviceHealthParameters"
    }
}
```

출력:

```
{
    "DeviceMethodResponse": "{\"remainingLife\": 99.8}"
}
```

자세한 내용은 AWS IoT 1-Click 개발자 안내서 AWS CLIAWS 의 에서 IoT 1-Click 사용을 참조하세요.

- 자세한 API 내용은 명령 참조InvokeDeviceMethod의 섹션을 참조하세요. AWS CLI

### list-device-events

다음 코드 예시에서는 list-device-events을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 시간 범위에 대한 디바이스의 이벤트를 나열하려면

다음 list-device-events 예제에서는 지정된 시간 범위에 대해 지정된 디바이스의 이벤트를 나열합니다.

```
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
```

출력:

```
{
    "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}"
        }
    ]
```

```
}
```

자세한 내용은 AWS IoT 1-Click 개발자 안내서 AWS CLIAWS 의 에서 IoT 1-Click 사용을 참조하세요.

- 자세한 API 내용은 명령 참조ListDeviceEvents의 섹션을 참조하세요. AWS CLI

### `list-devices`

다음 코드 예시에서는 `list-devices`을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 유형의 디바이스를 나열하려면

다음 `list-devices` 예제에서는 지정된 유형의 디바이스를 나열합니다.

```
aws iot1click-devices list-devices \
    --device-type button
```

이 명령은 출력을 생성하지 않습니다.

출력:

```
{
    "Devices": [
        {
            "remainingLife": 99.9,
            "attributes": {
                "arn": "arn:aws:iot1click:us-west-2:123456789012:devices/
G030PM0123456789",
                "type": "button",
                "deviceId": "G030PM0123456789",
                "enabled": false
            }
        }
    ]
}
```

자세한 내용은 AWS IoT 1-Click 개발자 안내서 AWS CLIAWS 의 에서 IoT 1-Click 사용을 참조하세요.

- 자세한 API 내용은 명령 참조ListDevices의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

디바이스의 태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 지정된 디바이스의 태그를 나열합니다.

```
aws iot1click-devices list-tags-for-resource \
    --resource-arn "arn:aws:iot1click:us-west-2:012345678901:devices/
G030PM0123456789"
```

출력:

```
{
    "Tags": {
        "Driver Phone": "123-555-0199",
        "Driver": "Jorge Souza"
    }
}
```

자세한 내용은 AWS IoT 1-Click 개발자 안내서 AWS CLIAWS 의 에서 IoT 1-Click 사용을 참조하세요.

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### `tag-resource`

다음 코드 예시에서는 `tag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

디바이스 AWS 리소스에 태그를 추가하려면

다음 `tag-resource` 예제에서는 지정된 리소스에 두 개의 태그를 추가합니다.

```
aws iot1click-devices tag-resource \
```

```
    --cli-input-json file://devices-tag-resource.json
```

devices-tag-resource.json의 콘텐츠:

```
{
    "ResourceArn": "arn:aws:iot1click:us-west-2:123456789012:devices/
G030PM0123456789",
    "Tags": {
        "Driver": "Jorge Souza",
        "Driver Phone": "123-555-0199"
    }
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 1-Click 개발자 안내서 AWS CLIAWS 의 에서 IoT 1-Click 사용을 참조하세요.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### unclaim-device

다음 코드 예시에서는 unclaim-device을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 계정에서 디바이스를 요청 취소(등록 취소)하려면

다음 unclaim-device 예제는 AWS 계정에서 지정된 디바이스를 요청 취소(등록 취소)합니다.

```
aws iot1click-devices unclaim-device \
    --device-id G030PM0123456789
```

출력:

```
{
    "State": "UNCLAIMED"
}
```

자세한 내용은 AWS IoT 1-Click 개발자 안내서 AWS CLIAWS 의 에서 IoT 1-Click 사용을 참조하세요.

- 자세한 API 내용은 명령 참조UnclaimDevice의 섹션을 참조하세요. AWS CLI

### `untag-resource`

다음 코드 예시에서는 `untag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

디바이스 AWS 리소스에서 태그를 제거하려면

다음 `untag-resource` 예제에서는 이름이 `Driver Phone` 및 인 태그를 지정된 디바이스 리소스`Driver`에서 제거합니다.

```
aws iot1click-devices untag-resource \
    --resource-arn "arn:aws:iot1click:us-west-2:123456789012:projects/
AnytownDumpsters" \
    --tag-keys "Driver Phone" "Driver"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 1-Click 개발자 안내서 AWS CLIAWS 의 에서 IoT 1-Click 사용을 참조하세요.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### `update-device-state`

다음 코드 예시에서는 `update-device-state`을 사용하는 방법을 보여 줍니다.

AWS CLI

디바이스의 ``enabled`` 상태를 업데이트하려면

다음은 지정된 디바이스의 상태를 로 `update-device-state` 설정합니다`enabled`.

```
aws iot1click-devices update-device-state \
    --device-id G030PM0123456789 \
    --enabled
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 1-Click 개발자 안내서 AWS CLIAWS 의 에서 IoT 1-Click 사용을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateDeviceState의 섹션을 참조하세요. AWS CLI

## AWS IoT 1-Click 를 사용한 프로젝트 예제 AWS CLI

다음 코드 예제에서는 AWS IoT 1-Click 프로젝트와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `associate-device-with-placement`

다음 코드 예시에서는 `associate-device-with-placement`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS IoT 1-Click 디바이스를 기존 배치와 연결하려면

다음 `associate-device-with-placement` 예제에서는 지정된 AWS IoT 1-Click 디바이스를 기존 배치와 연결합니다.

```
aws iot1click-projects associate-device-with-placement \
    --project-name AnytownDumpsters \
    --placement-name customer217 \
    --device-template-name empty-dumpster-request \
    --device-id G030PM0123456789
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 1-Click 개발자 안내서 AWS CLIAWS 의 에서 IoT 1-Click 사용을 참조하세요.

- 자세한 API 내용은 명령 참조AssociateDeviceWithPlacement의 섹션을 참조하세요. AWS CLI

### `create-placement`

다음 코드 예시에서는 `create-placement`을 사용하는 방법을 보여 줍니다.

AWS CLI

프로젝트의 AWS IoT 1-Click 배치를 생성하려면

다음 `create-placement` 예제에서는 지정된 프로젝트에 대한 AWS IoT 1-Click 배치를 생성합니다.

```
aws iot1click-projects create-placement \
    --project-name AnytownDumpsters \
    --placement-name customer217 \
    --attributes "{"location": "123 Any Street Anytown, USA 10001", "phone":
 "123-456-7890"}"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 1-Click 개발자 안내서 AWS CLIAWS 의 에서 IoT 1-Click 사용을 참조하세요.

- 자세한 API 내용은 명령 참조CreatePlacement의 섹션을 참조하세요. AWS CLI

### `create-project`

다음 코드 예시에서는 `create-project`을 사용하는 방법을 보여 줍니다.

AWS CLI

0개 이상의 배치에 대한 AWS IoT 1-Click 프로젝트를 생성하려면

다음 `create-project` 예제에서는 배치를 위한 AWS IoT 1-Click 프로젝트를 생성합니다.

aws iot1click-projects create-project --cli-input-json file://create-project.json

`create-project.json`의 콘텐츠:

```
{
     "projectName": "AnytownDumpsters",
     "description": "All dumpsters in the Anytown region.",
     "placementTemplate": {
         "defaultAttributes": {
             "City" : "Anytown"
         },
         "deviceTemplates": {
             "empty-dumpster-request" : {
                 "deviceType": "button"
             }
         }
     }
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 1-Click 개발자 안내서 AWS CLIAWS 의 에서 IoT 1-Click 사용을 참조하세요.

- 자세한 API 내용은 명령 참조CreateProject의 섹션을 참조하세요. AWS CLI

### `delete-placement`

다음 코드 예시에서는 `delete-placement`을 사용하는 방법을 보여 줍니다.

AWS CLI

프로젝트에서 배치를 삭제하려면

다음 `delete-placement` 예제에서는 프로젝트에서 지정된 배치를 삭제합니다.

```
aws iot1click-projects delete-placement \
    --project-name AnytownDumpsters \
    --placement-name customer217
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 1-Click 개발자 안내서 AWS CLIAWS 의 에서 IoT 1-Click 사용을 참조하세요.

- 자세한 API 내용은 명령 참조DeletePlacement의 섹션을 참조하세요. AWS CLI

### delete-project

다음 코드 예시에서는 delete-project을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 계정에서 프로젝트를 삭제하려면

다음 delete-project 예제에서는 AWS 계정에서 지정된 프로젝트를 삭제합니다.

```
aws iot1click-projects delete-project \
    --project-name AnytownDumpsters
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 1-Click 개발자 안내서 AWS CLIAWS 의 에서 IoT 1-Click 사용을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteProject의 섹션을 참조하세요. AWS CLI

### describe-placement

다음 코드 예시에서는 describe-placement을 사용하는 방법을 보여 줍니다.

AWS CLI

프로젝트의 배치를 설명하려면

다음 describe-placement 예제에서는 지정된 프로젝트의 배치를 설명합니다.

```
aws iot1click-projects describe-placement \
    --project-name AnytownDumpsters \
    --placement-name customer217
```

출력:

```
{
    "placement": {
        "projectName": "AnytownDumpsters",
        "placementName": "customer217",
        "attributes": {
            "phone": "123-555-0110",
            "location": "123 Any Street Anytown, USA 10001"
```

```
        },
        "createdDate": 1563488454,
        "updatedDate": 1563488454
    }
}
```

자세한 내용은 AWS IoT 1-Click 개발자 안내서 AWS CLIAWS 의 에서 IoT 1-Click 사용을 참조하세요.

- 자세한 API 내용은 명령 참조DescribePlacement의 섹션을 참조하세요. AWS CLI

### describe-project

다음 코드 예시에서는 describe-project을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS IoT 1-Click 프로젝트를 설명하려면

다음 describe-project 예제에서는 지정된 AWS IoT 1-Click 프로젝트를 설명합니다.

```
aws iot1click-projects describe-project \
    --project-name AnytownDumpsters
```

출력:

```
{
    "project": {
        "arn": "arn:aws:iot1click:us-west-2:012345678901:projects/AnytownDumpsters",
        "projectName": "AnytownDumpsters",
        "description": "All dumpsters in the Anytown region.",
        "createdDate": 1563483100,
        "updatedDate": 1563483100,
        "placementTemplate": {
            "defaultAttributes": {
                "City": "Anytown"
            },
            "deviceTemplates": {
                "empty-dumpster-request": {
                    "deviceType": "button",
                    "callbackOverrides": {}
                }
```

```
            }
        },
        "tags": {}
    }
}
```

자세한 내용은 AWS IoT 1-Click 개발자 안내서 AWS CLIAWS 의 에서 IoT 1-Click 사용을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeProject의 섹션을 참조하세요. AWS CLI

### disassociate-device-from-placement

다음 코드 예시에서는 disassociate-device-from-placement을 사용하는 방법을 보여 줍니다.

AWS CLI

배치에서 디바이스를 연결 해제하려면

다음 disassociate-device-from-placement 예제에서는 지정된 디바이스를 배치에서 연결 해제합니다.

```
aws iot1click-projects disassociate-device-from-placement \
    --project-name AnytownDumpsters \
    --placement-name customer217 \
    --device-template-name empty-dumpster-request
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 1-Click 개발자 안내서 AWS CLIAWS 의 에서 IoT 1-Click 사용을 참조하세요.

- 자세한 API 내용은 명령 참조DisassociateDeviceFromPlacement의 섹션을 참조하세요. AWS CLI

### get-devices-in-placement

다음 코드 예시에서는 get-devices-in-placement을 사용하는 방법을 보여 줍니다.

AWS CLI

프로젝트에 포함된 배치의 모든 디바이스를 나열하려면

다음 get-devices-in-placement 예제에서는 지정된 프로젝트에 포함된 지정된 배치의 모든 디바이스를 나열합니다.

```
aws iot1click-projects get-devices-in-placement \
    --project-name AnytownDumpsters \
    --placement-name customer217
```

출력:

```
{
    "devices": {
        "empty-dumpster-request": "G030PM0123456789"
    }
}
```

자세한 내용은 AWS IoT 1-Click 개발자 안내서 AWS CLIAWS 의 에서 IoT 1-Click 사용을 참조하세요.

- 자세한 API 내용은 명령 참조GetDevicesInPlacement의 섹션을 참조하세요. AWS CLI

### list-placements

다음 코드 예시에서는 list-placements을 사용하는 방법을 보여 줍니다.

AWS CLI

프로젝트의 모든 AWS IoT 1-Click 배치를 나열하려면

다음 list-placements 예제에서는 지정된 프로젝트의 모든 AWS IoT 1-Click 배치를 나열합니다.

```
aws iot1click-projects list-placements \
    --project-name AnytownDumpsters
```

출력:

```
{
    "placements": [
        {
            "projectName": "AnytownDumpsters",
```

```
            "placementName": "customer217",
            "createdDate": 1563488454,
            "updatedDate": 1563488454
        }
    ]
}
```

자세한 내용은 AWS IoT 1-Click 개발자 안내서 AWS CLIAWS 의 에서 IoT 1-Click 사용을 참조하세요.

- 자세한 API 내용은 명령 참조ListPlacements의 섹션을 참조하세요. AWS CLI

### `list-projects`

다음 코드 예시에서는 `list-projects`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 AWS IoT 1-Click 프로젝트를 나열하려면

다음 `list-projects` 예제에서는 계정의 모든 AWS IoT 1-Click 프로젝트를 나열합니다.

```
aws iot1click-projects list-projects
```

출력:

```
{
    "projects": [
        {
            "arn": "arn:aws:iot1click:us-west-2:012345678901:projects/
AnytownDumpsters",
            "projectName": "AnytownDumpsters",
            "createdDate": 1563483100,
            "updatedDate": 1563483100,
            "tags": {}
        }
    ]
}
```

자세한 내용은 AWS IoT 1-Click 개발자 안내서 AWS CLIAWS 의 에서 IoT 1-Click 사용을 참조하세요.

- 자세한 API 내용은 명령 참조ListProjects의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

프로젝트 리소스의 태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 지정된 프로젝트 리소스의 태그를 나열합니다.

```
aws iot1click-projects list-tags-for-resource \
    --resource-arn "arn:aws:iot1click:us-west-2:123456789012:projects/
AnytownDumpsters"
```

출력:

```
{
    "tags": {
        "Manager": "Li Juan",
        "Account": "45215"
    }
}
```

자세한 내용은 AWS IoT 1-Click 개발자 안내서 AWS CLIAWS 의 에서 IoT 1-Click 사용을 참조하세요.

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### `tag-resource`

다음 코드 예시에서는 `tag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

프로젝트 리소스에 태그를 추가하려면

다음 `tag-resource` 예제에서는 지정된 프로젝트 리소스에 두 개의 태그를 추가합니다.

```
aws iot1click-projects tag-resource \
```

```
    --cli-input-json file://devices-tag-resource.json
```

devices-tag-resource.json의 콘텐츠:

```
{
    "resourceArn": "arn:aws:iot1click:us-west-2:123456789012:projects/
AnytownDumpsters",
    "tags": {
        "Account": "45215",
        "Manager": "Li Juan"
    }
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 1-Click 개발자 안내서 AWS CLIAWS 의 에서 IoT 1-Click 사용을 참조하세요.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### untag-resource

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

프로젝트 리소스에서 태그를 제거하려면

다음 untag-resource 예제에서는 지정된 프로젝트Manager에서 키 이름이 인 태그를 제거합니다.

```
aws iot1click-projects untag-resource \
    --resource-arn "arn:aws:iot1click:us-west-2:123456789012:projects/
AnytownDumpsters" \
    --tag-keys "Manager"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 1-Click 개발자 안내서 AWS CLIAWS 의 에서 IoT 1-Click 사용을 참조하세요.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-placement

다음 코드 예시에서는 update-placement을 사용하는 방법을 보여 줍니다.

AWS CLI

배치의 '속성' 키-값 페어를 업데이트하려면

다음 update-placement 예제에서는 배치의 '속성' 키-값 페어를 업데이트합니다.

```
aws iot1click-projects update-placement \
    --cli-input-json file://update-placement.json
```

update-placement.json의 콘텐츠:

```
{
    "projectName": "AnytownDumpsters",
    "placementName": "customer217",
    "attributes": {
        "phone": "123-456-7890",
        "location": "123 Any Street Anytown, USA 10001"
    }
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 1-Click 개발자 안내서 AWS CLIAWS 의 에서 IoT 1-Click 사용을 참조하세요.

- 자세한 API 내용은 명령 참조UpdatePlacement의 섹션을 참조하세요. AWS CLI

### update-project

다음 코드 예시에서는 update-project을 사용하는 방법을 보여 줍니다.

AWS CLI

프로젝트의 설정을 업데이트하려면

다음 update-project 예제에서는 프로젝트에 대한 설명을 업데이트합니다.

```
aws iot1click-projects update-project \
```

```
    --project-name AnytownDumpsters \
    --description "All dumpsters (yard waste, recycling, garbage) in the Anytown
region."
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 1-Click 개발자 안내서 AWS CLIAWS 의 에서 IoT 1-Click 사용을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateProject의 섹션을 참조하세요. AWS CLI

## AWS IoT Analytics 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS IoT Analytics.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `batch-put-message`

다음 코드 예시에서는 `batch-put-message`을 사용하는 방법을 보여 줍니다.

AWS CLI

채널에 메시지를 보내려면

다음 `batch-put-message` 예제에서는 지정된 채널로 메시지를 보냅니다.

```
aws iotanalytics batch-put-message \
    --cli-binary-format raw-in-base64-out \
```

```
    --cli-input-json file://batch-put-message.json
```

batch-put-message.json의 콘텐츠:

```
{
    "channelName": "mychannel",
    "messages": [
        {
            "messageId": "0001",
            "payload": "eyAidGVtcGVyYXR1cmUiOiAyMCB9"
        }
    ]
}
```

출력:

```
{
    "batchPutMessageErrorEntries": []
}
```

자세한 내용은 IoT Analytics 참조BatchPutMessage의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조BatchPutMessage의 섹션을 참조하세요. AWS CLI

### cancel-pipeline-reprocessing

다음 코드 예시에서는 cancel-pipeline-reprocessing을 사용하는 방법을 보여 줍니다.

AWS CLI

파이프라인을 통한 데이터 재처리를 취소하려면

다음 cancel-pipeline-reprocessing 예제에서는 지정된 파이프라인을 통한 데이터 재처리를 취소합니다.

```
aws iotanalytics cancel-pipeline-reprocessing \
    --pipeline-name mypipeline \
    --reprocessing-id "6ad2764f-fb13-4de3-b101-4e74af03b043"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IoT Analytics 참조CancelPipelineReprocessing의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조CancelPipelineReprocessing의 섹션을 참조하세요. AWS CLI

### `create-channel`

다음 코드 예시에서는 `create-channel`을 사용하는 방법을 보여 줍니다.

AWS CLI

채널을 생성하려면

다음 `create-channel` 예제에서는 지정된 구성을 사용하여 채널을 생성합니다. 채널은 MQTT 주제에서 데이터를 수집하고 처리되지 않은 원시 메시지를 보관한 후 파이프라인에 데이터를 게시합니다.

```
aws iotanalytics create-channel \
    --cli-input-json file://create-channel.json
```

`create-channel.json`의 콘텐츠:

```
{
    "channelName": "mychannel",
    "retentionPeriod": {
        "unlimited": true
    },
    "tags": [
        {
            "key": "Environment",
            "value": "Production"
        }
    ]
}
```

출력:

```
{
    "channelArn": "arn:aws:iotanalytics:us-west-2:123456789012:channel/mychannel",
    "channelName": "mychannel",
    "retentionPeriod": {
        "unlimited": true
    }
```

```
}
```

자세한 내용은 IoT Analytics 참조CreateChannel의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조CreateChannel의 섹션을 참조하세요. AWS CLI

### create-dataset-content

다음 코드 예시에서는 create-dataset-content을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 세트의 콘텐츠를 생성하려면

다음 create-dataset-content 예제에서는 queryAction (SQL쿼리) 또는 containerAction (컨테이셔닝된 애플리케이션 실행)을 적용하여 지정된 데이터 세트의 콘텐츠를 생성합니다.

```
aws iotanalytics create-dataset-content \
    --dataset-name mydataset
```

출력:

```
{
    "versionId": "d494b416-9850-4670-b885-ca22f1e89d62"
}
```

자세한 내용은 IoT Analytics 참조CreateDatasetContent의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조CreateDatasetContent의 섹션을 참조하세요. AWS CLI

### create-dataset

다음 코드 예시에서는 create-dataset을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 세트를 생성하려면

다음 create-dataset 예제에서는 데이터 세트를 생성합니다. 데이터 세트는 queryAction (SQL쿼리) 또는 containerAction (컨테이셔닝된 애플리케이션 실행)을 적용하여 데이터 스토어에서 검색된 데이터를 저장합니다. 이 작업은 데이터 세트 스켈레톤을 생성합니다. 를 호출

하여 수동으로 CreateDatasetContent 또는 지정한 에 따라 자동으로 데이터 세트를 채울 수 trigger 있습니다.

```
aws iotanalytics create-dataset \
    --cli-input-json file://create-dataset.json
```

create-dataset.json의 콘텐츠:

```
{
    "datasetName": "mydataset",
    "actions": [
        {
            "actionName": "myDatasetAction",
            "queryAction": {
                "sqlQuery": "SELECT * FROM mydatastore"
            }
        }
    ],
    "retentionPeriod": {
        "unlimited": true
    },
    "tags": [
        {
            "key": "Environment",
            "value": "Production"
        }
    ]
}
```

출력:

```
{
    "datasetName": "mydataset",
    "retentionPeriod": {
        "unlimited": true
    },
    "datasetArn": "arn:aws:iotanalytics:us-west-2:123456789012:dataset/mydataset"
}
```

자세한 내용은 IoT Analytics 참조CreateDataset의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조CreateDataset의 섹션을 참조하세요. AWS CLI

### create-datastore

다음 코드 예시에서는 create-datastore을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 스토어 생성

다음 create-datastore 예제에서는 메시지의 리포지토리인 데이터 스토어를 생성합니다.

```
aws iotanalytics create-datastore \
    --cli-input-json file://create-datastore.json
```

create-datastore.json의 콘텐츠:

```
{
    "datastoreName": "mydatastore",
    "retentionPeriod": {
        "numberOfDays": 90
    },
    "tags": [
        {
            "key": "Environment",
            "value": "Production"
        }
    ]
}
```

출력:

```
{
    "datastoreName": "mydatastore",
    "datastoreArn": "arn:aws:iotanalytics:us-west-2:123456789012:datastore/
mydatastore",
    "retentionPeriod": {
        "numberOfDays": 90,
        "unlimited": false
    }
}
```

자세한 내용은 IoT Analytics 참조CreateDatastore의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조CreateDatastore의 섹션을 참조하세요. AWS CLI

### create-pipeline

다음 코드 예시에서는 create-pipeline을 사용하는 방법을 보여 줍니다.

AWS CLI

IoT Analytics 파이프라인 생성

다음 create-pipeline 예제에서는 파이프라인을 생성합니다. 파이프라인은 채널로부터 메시지를 사용하고 사용자가 데이터 스토어에 저장하기 전에 메시지를 처리할 수 있도록 합니다. 채널과 데이터 스토어 활동을 모두 지정해야 하며, 선택적으로 pipelineActivities 배열에 최대 23개의 추가 활동을 지정해야 합니다.

```
aws iotanalytics create-pipeline \
    --cli-input-json file://create-pipeline.json
```

create-pipeline.json의 콘텐츠:

```
{
    "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"
        }
    ]
}
```

출력:

```
{
    "pipelineArn": "arn:aws:iotanalytics:us-west-2:123456789012:pipeline/
mypipeline",
    "pipelineName": "mypipeline"
}
```

자세한 내용은 IoT Analytics 참조CreatePipeline의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조CreatePipeline의 섹션을 참조하세요. AWS CLI

### `delete-channel`

다음 코드 예시에서는 `delete-channel`을 사용하는 방법을 보여 줍니다.

AWS CLI

IoT Analytics 채널 삭제

다음 `delete-channel` 예제에서는 지정된 채널을 삭제합니다.

```
aws iotanalytics delete-channel \
    --channel-name mychannel
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IoT Analytics 참조DeleteChannel의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조DeleteChannel의 섹션을 참조하세요. AWS CLI

### `delete-dataset-content`

다음 코드 예시에서는 `delete-dataset-content`을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 세트 콘텐츠를 삭제하려면

다음 `delete-dataset-content` 예제에서는 지정된 데이터 세트의 내용을 삭제합니다.

```
aws iotanalytics delete-dataset-content \
    --dataset-name mydataset
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IoT Analytics 참조DeleteDatasetContent의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조DeleteDatasetContent의 섹션을 참조하세요. AWS CLI

### `delete-dataset`

다음 코드 예시에서는 `delete-dataset`을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 세트를 삭제하려면

다음 `delete-dataset` 예제에서는 지정된 데이터 세트를 삭제합니다. 이 작업을 수행하기 전에 데이터 세트의 내용을 삭제할 필요는 없습니다.

```
aws iotanalytics delete-dataset \
    --dataset-name mydataset
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IoT Analytics 참조DeleteDataset의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조DeleteDataset의 섹션을 참조하세요. AWS CLI

### `delete-datastore`

다음 코드 예시에서는 `delete-datastore`을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 스토어 삭제

다음 `delete-datastore` 예제에서는 지정된 데이터 스토어를 삭제합니다.

```
aws iotanalytics delete-datastore \
    --datastore-name mydatastore
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IoT Analytics 참조DeleteDatastore의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조DeleteDatastore의 섹션을 참조하세요. AWS CLI

### delete-pipeline

다음 코드 예시에서는 delete-pipeline을 사용하는 방법을 보여 줍니다.

AWS CLI

파이프라인을 삭제하려면

다음 delete-pipeline 예제에서는 지정된 파이프라인을 삭제합니다.

```
aws iotanalytics delete-pipeline \
    --pipeline-name mypipeline
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IoT Analytics 참조DeletePipeline의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조DeletePipeline의 섹션을 참조하세요. AWS CLI

### describe-channel

다음 코드 예시에서는 describe-channel을 사용하는 방법을 보여 줍니다.

AWS CLI

채널에 대한 정보를 검색하려면

다음 describe-channel 예제에서는 지정된 채널에 대한 통계를 포함한 세부 정보를 표시합니다.

```
aws iotanalytics describe-channel \
    --channel-name mychannel \
    --include-statistics
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 IoT Analytics 참조DescribeChannel의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조DescribeChannel의 섹션을 참조하세요. AWS CLI

### `describe-dataset`

다음 코드 예시에서는 `describe-dataset`을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 세트에 대한 정보를 검색하려면

다음 `describe-dataset` 예제에서는 지정된 데이터 세트에 대한 세부 정보를 표시합니다.

```
aws iotanalytics describe-dataset \
    --dataset-name mydataset
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 IoT Analytics 참조DescribeDataset의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조DescribeDataset의 섹션을 참조하세요. AWS CLI

### `describe-datastore`

다음 코드 예시에서는 `describe-datastore`을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 스토어에 대한 정보를 검색하려면

다음 `describe-datastore` 예제에서는 지정된 데이터 스토어에 대한 통계를 포함한 세부 정보를 표시합니다.

```
aws iotanalytics describe-datastore \
    --datastore-name mydatastore \
    --include-statistics
```

출력:

```
{
    "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
        }
    }
}
```

자세한 내용은 IoT Analytics 참조DescribeDatastore의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조DescribeDatastore의 섹션을 참조하세요. AWS CLI

### `describe-logging-options`

다음 코드 예시에서는 `describe-logging-options`을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 로깅 옵션을 검색하려면

다음 `describe-logging-options` 예제에서는 현재 AWS IoT Analytics 로깅 옵션을 보여줍니다.

```
aws iotanalytics describe-logging-options
```

이 명령은 출력을 생성하지 않습니다. 출력:

```
{
    "loggingOptions": {
        "roleArn": "arn:aws:iam::123456789012:role/service-role/myIoTAnalyticsRole",
        "enabled": true,
```

```
        "level": "ERROR"
    }
}
```

자세한 내용은 IoT Analytics 참조DescribeLoggingOptions의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조DescribeLoggingOptions의 섹션을 참조하세요. AWS CLI

### describe-pipeline

다음 코드 예시에서는 describe-pipeline을 사용하는 방법을 보여 줍니다.

AWS CLI

파이프라인에 대한 정보를 검색하려면

다음 describe-pipeline 예제에서는 지정된 파이프라인에 대한 세부 정보를 표시합니다.

```
aws iotanalytics describe-pipeline \
    --pipeline-name mypipeline
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 IoT Analytics 참조DescribePipeline의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조DescribePipeline의 섹션을 참조하세요. AWS CLI

### get-dataset-content

다음 코드 예시에서는 get-dataset-content을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 세트의 내용을 검색하려면

다음 get-dataset-content 예제에서는 미리 서명된 데이터 세트의 내용을 검색합니다URIs.

```
aws iotanalytics get-dataset-content --dataset-name mydataset
```

출력:

```
{
    "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=..."
        }
```

```
    ]
}
```

자세한 내용은 가이드의 섹션을 참조GetDatasetContent하세요.

- 자세한 API 내용은 명령 참조GetDatasetContent의 섹션을 참조하세요. AWS CLI

### `list-channels`

다음 코드 예시에서는 `list-channels`을 사용하는 방법을 보여 줍니다.

AWS CLI

채널 목록을 검색하려면

다음 `list-channels` 예제에서는 사용 가능한 채널에 대한 요약 정보를 표시합니다.

```
aws iotanalytics list-channels
```

출력:

```
{
    "channelSummaries": [
        {
            "status": "ACTIVE",
            "channelName": "mychannel",
            "creationTime": 1557860351.001,
            "lastUpdateTime": 1557860351.001
        }
    ]
}
```

자세한 내용은 IoT Analytics 참조ListChannels의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조ListChannels의 섹션을 참조하세요. AWS CLI

### `list-dataset-contents`

다음 코드 예시에서는 `list-dataset-contents`을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 세트 콘텐츠에 대한 정보를 나열하려면

다음 `list-dataset-contents` 예제에서는 생성된 데이터 세트 콘텐츠에 대한 정보를 나열합니다.

```
aws iotanalytics list-dataset-contents \
    --dataset-name mydataset
```

출력:

```
{
    "datasetContentSummaries": [
        {
            "status": {
                "state": "SUCCEEDED"
            },
            "scheduleTime": 1557863215.995,
            "version": "b10ea2a9-66c1-4d99-8d1f-518113b738d0",
            "creationTime": 1557863215.995
        }
    ]
}
```

자세한 내용은 IoT Analytics 참조ListDatasetContents의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조ListDatasetContents의 섹션을 참조하세요. AWS CLI

### `list-datasets`

다음 코드 예시에서는 `list-datasets`을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 세트에 대한 정보를 검색하려면

다음 `list-datasets` 예제에서는 사용 가능한 데이터 세트에 대한 요약 정보를 나열합니다.

```
aws iotanalytics list-datasets
```

출력:

```
{
    "datasetSummaries": [
```

```
        {
            "status": "ACTIVE",
            "datasetName": "mydataset",
            "lastUpdateTime": 1557859240.658,
            "triggers": [],
            "creationTime": 1557859240.658,
            "actions": [
                {
                    "actionName": "query_32",
                    "actionType": "QUERY"
                }
            ]
        }
    ]
}
```

자세한 내용은 IoT Analytics 참조ListDatasets의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조ListDatasets의 섹션을 참조하세요. AWS CLI

### `list-datastores`

다음 코드 예시에서는 `list-datastores`을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 스토어 목록을 검색하려면

다음 `list-datastores` 예제에서는 사용 가능한 데이터 스토어에 대한 요약 정보를 표시합니다.

```
aws iotanalytics list-datastores
```

출력:

```
{
    "datastoreSummaries": [
        {
            "status": "ACTIVE",
            "datastoreName": "mydatastore",
            "creationTime": 1557858971.02,
            "lastUpdateTime": 1557858971.02
        }
    ]
```

```
}
```

자세한 내용은 IoT Analytics 참조ListDatastores의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조ListDatastores의 섹션을 참조하세요. AWS CLI

### `list-pipelines`

다음 코드 예시에서는 `list-pipelines`을 사용하는 방법을 보여 줍니다.

AWS CLI

파이프라인 목록을 검색하려면

다음 `list-pipelines` 예제에서는 사용 가능한 파이프라인 목록을 표시합니다.

```
aws iotanalytics list-pipelines
```

출력:

```
{
    "pipelineSummaries": [
        {
            "pipelineName": "mypipeline",
            "creationTime": 1557859124.432,
            "lastUpdateTime": 1557859124.432,
            "reprocessingSummaries": []
        }
    ]
}
```

자세한 내용은 IoT Analytics 참조ListPipelines의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조ListPipelines의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스의 태그를 나열하려면

다음 list-tags-for-resource 예제에서는 지정된 리소스에 연결한 태그를 나열합니다.

```
aws iotanalytics list-tags-for-resource \
    --resource-arn "arn:aws:iotanalytics:us-west-2:123456789012:channel/mychannel"
```

출력:

```
{
    "tags": [
        {
            "value": "bar",
            "key": "foo"
        }
    ]
}
```

자세한 내용은 IoT Analytics 참조ListTagsForResource의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### put-logging-options

다음 코드 예시에서는 put-logging-options을 사용하는 방법을 보여 줍니다.

AWS CLI

로깅 옵션을 설정하거나 업데이트하려면

다음 put-logging-options 예제에서는 AWS IoT Analytics 로깅 옵션을 설정하거나 업데이트합니다. loggingOptions 필드의 값을 업데이트하면 변경 사항이 적용되는 데 최대 1분이 걸릴 수 있습니다. 또한 “roleArn” 필드에서 지정한 역할에 연결된 정책을 변경하는 경우(예: 잘못된 정책을 수정하는 경우) 해당 변경 사항이 적용되는 데 최대 5분이 걸릴 수 있습니다.

```
aws iotanalytics put-logging-options \
    --cli-input-json file://put-logging-options.json
```

put-logging-options.json의 콘텐츠:

```
{
    "loggingOptions": {
        "roleArn": "arn:aws:iam::123456789012:role/service-role/myIoTAnalyticsRole",
```

```
        "level": "ERROR",
        "enabled": true
    }
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IoT Analytics 참조PutLoggingOptions의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조PutLoggingOptions의 섹션을 참조하세요. AWS CLI

### `run-pipeline-activity`

다음 코드 예시에서는 `run-pipeline-activity`을 사용하는 방법을 보여 줍니다.

AWS CLI

파이프라인 활동을 시뮬레이션하려면

다음 `run-pipeline-activity` 예제에서는 메시지 페이로드에서 파이프라인 활동을 실행한 결과를 시뮬레이션합니다.

```
aws iotanalytics run-pipeline-activity \
    --pipeline-activity file://maths.json \
    --payloads file://payloads.json
```

`maths.json`의 콘텐츠:

```
{
    "math": {
        "name": "MyMathActivity",
        "math": "((temp - 32) * 5.0) / 9.0",
        "attribute": "tempC"
    }
}
```

`payloads.json`의 콘텐츠:

```
[
    "{\"humidity\": 52, \"temp\": 68 }",
    "{\"humidity\": 52, \"temp\": 32 }"
```

```
]
```

출력:

```
{
    "logResult": "",
    "payloads": [
        "eyJodW1pZGl0eSI6NTIsInRlbXAiOjY4LCJ0ZW1wQyI6MjB9",
        "eyJodW1pZGl0eSI6NTIsInRlbXAiOjMyLCJ0ZW1wQyI6MH0="
    ]
}
```

자세한 내용은 IoT Analytics 참조RunPipelineActivity의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조RunPipelineActivity의 섹션을 참조하세요. AWS CLI

### `sample-channel-data`

다음 코드 예시에서는 `sample-channel-data`을 사용하는 방법을 보여 줍니다.

AWS CLI

채널에서 샘플 메시지를 검색하려면

다음 `sample-channel-data` 예제에서는 지정된 기간 동안 수집된 지정된 채널에서 메시지 샘플을 검색합니다. 최대 10개의 메시지를 검색할 수 있습니다.

```
aws iotanalytics sample-channel-data \
    --channel-name mychannel
```

출력:

```
{
    "payloads": [
        "eyAidGVtcGVyYXR1cmUiOiAyMCB9",
        "eyAiZm9vIjogImJhciIgfQ=="
    ]
}
```

자세한 내용은 IoT Analytics 참조SampleChannelData의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조SampleChannelData의 섹션을 참조하세요. AWS CLI

### start-pipeline-reprocessing

다음 코드 예시에서는 start-pipeline-reprocessing을 사용하는 방법을 보여 줍니다.

AWS CLI

파이프라인 재처리를 시작하려면

다음 start-pipeline-reprocessing 예제에서는 지정된 파이프라인을 통해 원시 메시지 데이터의 재처리를 시작합니다.

```
aws iotanalytics start-pipeline-reprocessing \
    --pipeline-name mypipeline
```

출력:

```
{
    "reprocessingId": "6ad2764f-fb13-4de3-b101-4e74af03b043"
}
```

자세한 내용은 IoT Analytics 참조StartPipelineReprocessing의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조StartPipelineReprocessing의 섹션을 참조하세요. AWS CLI

### tag-resource

다음 코드 예시에서는 tag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 대한 태그를 추가하거나 수정하려면

다음 tag-resource 예제에서는 지정된 리소스에 연결된 태그를 추가하거나 수정합니다.

```
aws iotanalytics tag-resource \
    --resource-arn "arn:aws:iotanalytics:us-west-2:123456789012:channel/mychannel" \
    --tags "[{\"key\": \"Environment\", \"value\": \"Production\"}]"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IoT Analytics 참조TagResource의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### untag-resource

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에서 태그를 제거하려면

다음 untag-resource 예제에서는 지정된 리소스에서 지정된 키 이름이 있는 태그를 제거합니다.

```
aws iotanalytics untag-resource \
    --resource-arn "arn:aws:iotanalytics:us-west-2:123456789012:channel/mychannel" \
    --tag-keys "[\"Environment\"]"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT Analytics API 참조의 UntagResource <https://docs.aws.amazon.com/iotanalytics/latest/APIReference/API_UntagResource.html >을 참조하세요.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-channel

다음 코드 예시에서는 update-channel을 사용하는 방법을 보여 줍니다.

AWS CLI

채널을 수정하려면

다음 update-channel 예제에서는 지정된 채널에 대한 설정을 수정합니다.

```
aws iotanalytics update-channel \
    --cli-input-json file://update-channel.json
```

update-channel.json의 콘텐츠:

```
{
    "channelName": "mychannel",
    "retentionPeriod": {
        "numberOfDays": 92
    }
```

```
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IoT Analytics 참조UpdateChannel의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조UpdateChannel의 섹션을 참조하세요. AWS CLI

### update-dataset

다음 코드 예시에서는 update-dataset을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 세트를 업데이트하려면

다음 update-dataset 예제에서는 지정된 데이터 세트의 설정을 수정합니다.

```
aws iotanalytics update-dataset \
    --cli-input-json file://update-dataset.json
```

update-dataset.json의 콘텐츠:

```
{
    "datasetName": "mydataset",
    "actions": [
        {
            "actionName": "myDatasetUpdateAction",
            "queryAction": {
                "sqlQuery": "SELECT * FROM mydatastore"
            }
        }
    ],
    "retentionPeriod": {
        "numberOfDays": 92
    }
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT Analytics API 참조의 UpdateDataset <https://docs.aws.amazon.com/iotanalytics/latest/APIReference/API_UpdateDataset.html >을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateDataset의 섹션을 참조하세요. AWS CLI

### update-datastore

다음 코드 예시에서는 update-datastore을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 스토어를 업데이트하려면

다음 update-datastore 예제에서는 지정된 데이터 스토어의 설정을 수정합니다.

```
aws iotanalytics update-datastore \
    --cli-input-json file://update-datastore.json
```

update-datastore.json의 내용:

```
{
    "datastoreName": "mydatastore",
    "retentionPeriod": {
        "numberOfDays": 93
    }
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IoT Analytics 참조UpdateDatastore의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조UpdateDatastore의 섹션을 참조하세요. AWS CLI

### update-pipeline

다음 코드 예시에서는 update-pipeline을 사용하는 방법을 보여 줍니다.

AWS CLI

파이프라인을 업데이트하려면

다음 update-pipeline 예제에서는 지정된 파이프라인의 설정을 수정합니다. 채널과 데이터 스토어 활동을 모두 지정해야 하며, 선택적으로 pipelineActivities 배열에 최대 23개의 추가 활동을 지정해야 합니다.

```
aws iotanalytics update-pipeline \
    --cli-input-json file://update-pipeline.json
```

update-pipeline.json의 내용:

```
{
    "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"
            }
        }
    ]
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IoT Analytics 참조UpdatePipeline의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조UpdatePipeline의 섹션을 참조하세요. AWS CLI

## 를 사용한 Device Advisor 예제 AWS CLI

다음 코드 예제에서는 Device Advisor AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `create-suite-definition`

다음 코드 예시에서는 `create-suite-definition`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: IoT Device Advisor 테스트 제품군 생성

다음 `create-suite-definition` 예제에서는 지정된 제품군 정의 구성을 사용하여 AWS IoT에 Device Advisor 테스트 제품군을 생성합니다.

```
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"}'
```

출력:

```
{
    "suiteDefinitionId": "0jtsgio7yenu",
    "suiteDefinitionArn": "arn:aws:iotdeviceadvisor:us-
east-1:123456789012:suitedefinition/0jtsgio7yenu",
```

```
    "suiteDefinitionName": "TestSuiteName",
    "createdAt": "2022-12-02T11:38:13.263000-05:00"
}
```

자세한 내용은 AWS IoT Core 개발자 안내서의 테스트 제품군 정의 생성을 참조하세요.

예제 2: IoT Device Advisor 최신 자격 테스트 제품군 생성

다음 `create-suite-definition` 예제에서는 지정된 제품군 정의 구성으로 AWS IoT의 최신 버전을 사용하여 디바이스 어드바이저 자격 테스트 제품군을 생성합니다.

```
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"}'
```

출력:

```
{
    "suiteDefinitionId": "txgsuolk2myj",
    "suiteDefinitionArn": "arn:aws:iotdeviceadvisor:us-
east-1:123456789012:suitedefinition/txgsuolk2myj",
    "suiteDefinitionName": "TestSuiteName",
    "createdAt": "2022-12-02T11:38:13.263000-05:00"
}
```

자세한 내용은 AWS IoT Core 개발자 안내서의 테스트 제품군 정의 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateSuiteDefinition의 섹션을 참조하세요. AWS CLI

### `delete-suite-definition`

다음 코드 예시에서는 `delete-suite-definition`을 사용하는 방법을 보여 줍니다.

AWS CLI

IoT Device Advisor 테스트 제품군을 삭제하려면

다음 delete-suite-definition 예제에서는 지정된 제품군 정의 ID가 있는 디바이스 어드바이저 테스트 제품군을 삭제합니다.

```
aws iotdeviceadvisor delete-suite-definition \
    --suite-definition-id 0jtsgio7yenu
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IoT 참조DeleteSuiteDefinition의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조DeleteSuiteDefinition의 섹션을 참조하세요. AWS CLI

### get-endpoint

다음 코드 예시에서는 get-endpoint을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: IoT Device Advisor 계정 수준 엔드포인트에 대한 정보를 가져오려면

다음 get-endpoint 예제에서는 디바이스 어드바이저 계정 수준 테스트 엔드포인트에 대한 정보를 가져옵니다.

```
aws iotdeviceadvisor get-endpoint
```

출력:

```
{
    "endpoint": "t6y4c143x9sfo.deviceadvisor.iot.us-east-1.amazonaws.com"
}
```

예제 2: IoT Device Advisor 디바이스 수준 엔드포인트에 대한 정보를 가져오려면

다음 get-endpoint 예제에서는 지정된 사물 배열 또는 인증서 배열을 사용하는 디바이스 어드바이저 디바이스 수준 테스트 엔드포인트에 대한 정보를 가져옵니다.

```
aws iotdeviceadvisor get-endpoint \
    --thing-arn arn:aws:iot:us-east-1:123456789012:thing/MyIotThing
```

출력:

```
{
    "endpoint": "tdb7719be5t6y4c143x9sfo.deviceadvisor.iot.us-east-1.amazonaws.com"
}
```

자세한 내용은 AWS IoT Core 개발자 안내서의 테스트 엔드포인트 가져오기를 참조하세요.

- 자세한 API 내용은 명령 참조GetEndpoint의 섹션을 참조하세요. AWS CLI

### get-suite-definition

다음 코드 예시에서는 get-suite-definition을 사용하는 방법을 보여 줍니다.

AWS CLI

IoT Device Advisor 테스트 제품군에 대한 정보를 가져오려면

다음 get-suite-definition 예제에서는 지정된 제품군 정의 ID가 있는 Aevice Advisor 테스트 제품군에 대한 정보를 가져옵니다.

```
aws iotdeviceadvisor get-suite-definition \
    --suite-definition-id qqcsmtyyjabl
```

출력:

```
{
    "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": {}
}
```

자세한 내용은 AWS IoT Core 개발자 안내서의 테스트 제품군 정의 가져오기를 참조하세요.

- 자세한 API 내용은 명령 참조GetSuiteDefinition의 섹션을 참조하세요. AWS CLI

### get-suite-run-report

다음 코드 예시에서는 get-suite-run-report을 사용하는 방법을 보여 줍니다.

AWS CLI

IoT Device Advisor 적격 테스트 제품군 실행 보고서에 대한 정보를 가져오려면

다음 get-suite-run-report 예제에서는 지정된 제품군 정의 ID 및 제품군 실행 ID로 성공적인 디바이스 어드바이저 자격 테스트 제품군 실행에 대한 보고서 다운로드 링크를 가져옵니다.

```
aws iotdeviceadvisor get-suite-run-report \
    --suite-definition-id ztvb5aek4w4x \
    --suite-run-id p6awv83nre6v
```

출력:

```
{
    "qualificationReportDownloadUrl": "https://senate-apn-reports-us-east-1-
prod.s3.amazonaws.com/report.downloadlink"
}
```

자세한 내용은 AWS IoT Core 개발자 안내서의 성공적인 자격 테스트 제품군 실행을 위한 자격 보고서 가져오기를 참조하세요.

- 자세한 API 내용은 명령 참조GetSuiteRunReport의 섹션을 참조하세요. AWS CLI

### get-suite-run

다음 코드 예시에서는 get-suite-run을 사용하는 방법을 보여 줍니다.

AWS CLI

IoT Device Advisor 테스트 제품군 실행 상태에 대한 정보를 가져오려면

다음 `get-suite-run` 예제에서는 지정된 제품군 정의 ID 및 제품군 실행 ID를 사용하여 디바이스 어드바이저 테스트 제품군 실행 상태에 대한 정보를 가져옵니다.

```
aws iotdeviceadvisor get-suite-run \
    --suite-definition-id qqcsmtyyjabl \
    --suite-run-id nzlfyhaa18oa
```

출력:

```
{
    "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": {}
}
```

자세한 내용은 AWS IoT Core 개발자 안내서의 테스트 제품군 실행 가져오기를 참조하세요.

- 자세한 API 내용은 명령 참조GetSuiteRun의 섹션을 참조하세요. AWS CLI

### `list-suite-definitions`

다음 코드 예시에서는 `list-suite-definitions`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 생성한 IoT Device Advisor 테스트 제품군을 나열하려면

다음 `list-suite-definitions` 예제에서는 AWS IoT 에서 생성한 최대 25개의 디바이스 어드바이저 테스트 제품군을 나열합니다. 테스트 제품군이 25개 이상인 경우 출력에 “nextToken”가 표시됩니다. 이 “nextToken”를 사용하여 생성한 나머지 테스트 제품군을 표시할 수 있습니다.

```
aws iotdeviceadvisor list-suite-definitions
```

출력:

```
{
    "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"
}
```

예제 2: 지정된 설정으로 생성한 IoT Device Advisor 테스트 제품군을 나열하려면

다음 `list-suite-definitions` 예제에서는 AWS IoT에서 생성한 디바이스 어드바이저 테스트 제품군을 지정된 최대 결과 번호와 함께 나열합니다. 최대 수보다 더 많은 테스트 제품군이 있는 경우 출력에 “nextToken”가 표시됩니다. “nextToken”가 있는 경우 “nextToken”를 사용하여 이전에 표시되지 않은 생성한 테스트 제품군을 표시할 수 있습니다.

```
aws iotdeviceadvisor list-suite-definitions \
    --max-result 1 \
    --next-token "nextTokenValue"
```

출력:

```
{
    "suiteDefinitionInformationList": [
        {
            "suiteDefinitionId": "ztvb5aew4w4x",
            "suiteDefinitionName": "TestSuite2",
            "defaultDevices": [],
            "intendedForQualification": true,
            "isLongDurationTest": false,
            "protocol": "MqttV3_1_1",
            "createdAt": "2022-11-17T14:15:56.830000-05:00"
        }
    ],
    "nextToken": "nextTokenValue"
}
```

자세한 내용은 IoT 참조ListSuiteDefinitions의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조ListSuiteDefinitions의 섹션을 참조하세요. AWS CLI

### `list-suite-runs`

다음 코드 예시에서는 `list-suite-runs`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 지정된 IoT Device Advisor 테스트 제품군 실행 상태에 대한 모든 정보를 나열하려면

다음 `list-suite-runs` 예제에서는 디바이스 어드바이저 테스트 제품군 실행 상태에 대한 모든 정보를 지정된 제품군 정의 ID와 함께 나열합니다. 테스트 제품군이 25개 이상 실행되면 출력에 “nextToken”가 표시됩니다. 이 “nextToken”를 사용하여 나머지 테스트 제품군 실행을 표시할 수 있습니다.

```
aws iotdeviceadvisor list-suite-runs \
    --suite-definition-id ztvb5aew4w4x
```

출력:

```
{
    "suiteRunsList": [
        {
            "suiteDefinitionId": "ztvb5aew4w4x",
            "suiteDefinitionVersion": "v1",
            "suiteDefinitionName": "TestSuite",
            "suiteRunId": "p6awv89nre6v",
            "createdAt": "2022-12-01T16:33:14.212000-05:00",
            "startedAt": "2022-12-01T16:33:15.710000-05:00",
            "endAt": "2022-12-01T16:42:03.323000-05:00",
            "status": "PASS",
            "passed": 6,
            "failed": 0
        }
    ]
}
```

예제 2: 지정된 IoT Device Advisor 테스트 제품군 실행 상태에 대한 정보를 지정된 설정으로 나열하려면

다음 `list-suite-runs` 예제에서는 디바이스 어드바이저 테스트 제품군 실행 상태에 대한 정보와 지정된 제품군 정의 ID 및 지정된 최대 결과 번호를 나열합니다. 최댓값보다 더 많은 테스트 제품

군 실행이 있는 경우 출력에 “nextToken”가 표시됩니다. “nextToken”가 있는 경우 “nextToken”를 사용하여 이전에 표시되지 않은 테스트 제품군 실행을 표시할 수 있습니다.

```
aws iotdeviceadvisor list-suite-runs \
    --suite-definition-id qqcsmtyyjaml \
    --max-result 1 \
    --next-token "nextTokenValue"
```

출력:

```
{
    "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"
}
```

자세한 내용은 IoT 참조ListSuiteRuns의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조ListSuiteRuns의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

IoT Device Advisor 리소스에 연결된 태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 디바이스 어드바이저 리소스에 연결된 태그를 나열합니다. 디바이스 어드바이저 리소스는 Suitedefinition-Arn 또는 Suiterun-Arn일 수 있습니다.

```
aws iotdeviceadvisor list-tags-for-resource \
    --resource-arn arn:aws:iotdeviceadvisor:us-east-1:123456789012:suitedefinition/
ba0uyjpg38ny
```

출력:

```
{
    "tags": {
        "TestTagKey": "TestTagValue"
    }
}
```

자세한 내용은 서비스 승인 참조ListTagsForResource의 IoT Core Device Advisor에서 정의한 AWS IoT 참조 및 리소스 유형의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### start-suite-run

다음 코드 예시에서는 start-suite-run을 사용하는 방법을 보여 줍니다.

AWS CLI

IoT Device Advisor 테스트 제품군 실행을 시작하려면

다음 start-suite-run 예제에서는 AWS 계정에서 사용 가능한 위젯을 나열합니다.

```
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"}}'
```

출력:

```
{
    "suiteRunId": "pwmucgw7lt9s",
    "suiteRunArn": "arn:aws:iotdeviceadvisor:us-east-1:123456789012:suiterun/
qqcsmtyyjabl/pwmucgw7lk9s",
```

```
    "createdAt": "2022-12-02T15:43:05.581000-05:00"
}
```

자세한 내용은 AWS IoT Core 개발자 안내서의 테스트 제품군 실행 시작을 참조하세요.

- 자세한 API 내용은 명령 참조StartSuiteRun의 섹션을 참조하세요. AWS CLI

### stop-suite-run

다음 코드 예시에서는 stop-suite-run을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 실행 중인 IoT Device Advisor 테스트 제품군을 중지하려면

다음 stop-suite-run 예제에서는 현재 지정된 제품군 정의 ID 및 제품군 실행 ID로 실행 중인 디바이스 어드바이저 테스트 제품군을 중지합니다.

```
aws iotdeviceadvisor stop-suite-run \
    --suite-definition-id qqcsmtyyjabl \
    --suite-run-id nzlfyhaa18oa
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT Core 개발자 안내서의 테스트 제품군 실행 중지를 참조하세요.

- 자세한 API 내용은 명령 참조StopSuiteRun의 섹션을 참조하세요. AWS CLI

### tag-resource

다음 코드 예시에서는 tag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

IoT Device Advisor 리소스의 기존 태그를 에 추가하고 수정하려면

다음 tag-resource 예제에서는 지정된 리소스 arn 및 태그를 사용하여 디바이스 어드바이저 리소스의 기존 태그를 추가하고 수정합니다. 디바이스 어드바이저 리소스는 Suitedefinition-Arn 또는 Suiterun-Arn일 수 있습니다.

```
aws iotdeviceadvisor tag-resource \
```

```
    --resource-arn arn:aws:iotdeviceadvisor:us-east-1:123456789012:suitedefinition/
ba0uyjpg38ny \
    --tags '{"TagKey": "TagValue"}'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 서비스 승인 참조TagResource의 IoT Core Device Advisor에서 정의한 AWS IoT 참조 및 리소스 유형의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### untag-resource

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

IoT Device Advisor 리소스에서 기존 태그를 제거하려면

다음 untag-resource 예제에서는 지정된 리소스 arn 및 태그 키를 사용하여 디바이스 어드바이저 리소스에서 기존 태그를 제거합니다. 디바이스 어드바이저 리소스는 Suitedefinition-Arn 또는 Suiterun-Arn일 수 있습니다.

```
aws iotdeviceadvisor untag-resource \
    --resource-arn arn:aws:iotdeviceadvisor:us-east-1:123456789012:suitedefinition/
ba0uyjpg38ny \
    --tag-keys "TagKey"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 서비스 승인 참조UntagResource의 IoT Core Device Advisor에서 정의한 AWS IoT 참조 및 리소스 유형의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-suite-definition

다음 코드 예시에서는 update-suite-definition을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: IoT Device Advisor 테스트 제품군 업데이트

다음 update-suite-definition 예제에서는 AWS IoT의 디바이스 어드바이저 테스트 제품군을 지정된 제품군 정의 ID 및 제품군 정의 구성으로 업데이트합니다.

```
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"}'
```

출력:

```
{
    "suiteDefinitionId": "3hsn88h4p2g5",
    "suiteDefinitionName": "TestSuiteName",
    "suiteDefinitionVersion": "v3",
    "createdAt": "2022-11-17T14:15:56.830000-05:00",
    "lastUpdatedAt": "2022-12-02T16:02:45.857000-05:00"
}
```

예제 2: IoT Device Advisor 자격 테스트 제품군 업데이트

다음 update-suite-definition 예제에서는 AWS IoT의 디바이스 어드바이저 자격 테스트 제품군을 지정된 제품군 정의 ID 및 제품군 정의 구성으로 업데이트합니다.

```
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"}'
```

출력:

```
{
    "suiteDefinitionId": "txgsuolk2myj",
    "suiteDefinitionName": "TestSuiteName",
    "suiteDefinitionVersion": "v3",
    "createdAt": "2022-11-17T14:15:56.830000-05:00",
    "lastUpdatedAt": "2022-12-02T16:02:45.857000-05:00"
}
```

자세한 내용은 IoT 참조UpdateSuiteDefinition의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조UpdateSuiteDefinition의 섹션을 참조하세요. AWS CLI

## AWS IoT data 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS IoT data.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `delete-thing-shadow`

다음 코드 예시에서는 `delete-thing-shadow`을 사용하는 방법을 보여 줍니다.

AWS CLI

디바이스의 섀도우 문서를 삭제하려면

다음 `delete-thing-shadow` 예제에서는 이름이 인 디바이스의 전체 섀도우 문서를 삭제합니다`MyRPi`.

```
aws iot-data delete-thing-shadow \
    --thing-name MyRPi \
    "output.txt"
```

이 명령은 디스플레이에 출력을 생성하지 않지만 삭제한 섀도우 문서의 버전 및 타임스탬프를 확인하는 정보가 output.txt 포함되어 있습니다.

```
{"version":2,"timestamp":1560270384}
```

자세한 내용은 IoT 개발자 안내서의 섀도우 사용을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조DeleteThingShadow의 섹션을 참조하세요. AWS CLI

### get-thing-shadow

다음 코드 예시에서는 get-thing-shadow을 사용하는 방법을 보여 줍니다.

AWS CLI

사물 섀도우 문서를 가져오려면

다음 get-thing-shadow 예제에서는 지정된 IoT 사물에 대한 사물 섀도우 문서를 가져옵니다.

```
aws iot-data get-thing-shadow \
    --thing-name MyRPi \
    output.txt
```

이 명령은 디스플레이에 출력을 생성하지 않지만, 다음은 의 내용을 보여줍니다output.txt.

```
{
  "state":{
    "reported":{
    "moisture":"low"
    }
  },
  "metadata":{
    "reported":{
      "moisture":{
        "timestamp":1560269319
      }
```

```
    }
  },
  "version":1,"timestamp":1560269405
}
```

자세한 내용은 IoT 개발자 안내서의 Device Shadow Service Data Flow를 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조GetThingShadow의 섹션을 참조하세요. AWS CLI

### update-thing-shadow

다음 코드 예시에서는 update-thing-shadow을 사용하는 방법을 보여 줍니다.

AWS CLI

사물 섀도우를 업데이트하려면

다음 update-thing-shadow 예제에서는 지정된 사물에 대한 디바이스 섀도우의 현재 상태를 수정하고 파일에 저장합니다output.txt.

```
aws iot-data update-thing-shadow \
    --thing-name MyRPi \
    --payload "{"state":{"reported":{"moisture":"okay"}}}" \
    "output.txt"
```

이 명령은 디스플레이에 출력을 생성하지 않지만, 다음은 의 내용을 보여줍니다output.txt.

```
{
    "state": {
        "reported": {
            "moisture": "okay"
        }
    },
    "metadata": {
        "reported": {
            "moisture": {
                "timestamp": 1560270036
            }
        }
    },
    "version": 2,
    "timestamp": 1560270036
```

```
}
```

자세한 내용은 IoT 개발자 안내서의 Device Shadow Service Data Flow를 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조UpdateThingShadow의 섹션을 참조하세요. AWS CLI

## AWS IoT Events 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS IoT Events.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `batch-put-message`

다음 코드 예시에서는 `batch-put-message`을 사용하는 방법을 보여 줍니다.

AWS CLI

메시지(입력)를 AWS IoT Events로 보내려면

다음 `batch-put-message` 예제에서는 AWS IoT Events 시스템에 메시지 세트를 전송합니다. 각 메시지 페이로드는 지정한 입력(`inputName`)으로 변환되고 해당 입력을 모니터링하는 모든 탐지기로 수집됩니다. 여러 메시지가 전송되는 경우 메시지가 처리되는 순서가 보장되지 않습니다. 주문을 보장하려면 메시지를 한 번에 하나씩 보내고 응답이 성공할 때까지 기다려야 합니다.

```
aws iotevents-data batch-put-message \
    --cli-input-json file://highPressureMessage.json
```

`highPressureMessage.json`의 콘텐츠:

```
{
    "messages": [
        {
            "messageId": "00001",
            "inputName": "PressureInput",
            "payload": "{\"motorid\": \"Fulton-A32\", \"sensorData\": {\"pressure\":
 80, \"temperature\": 39} }"
        }
    ]
}
```

출력:

```
{
    "BatchPutMessageErrorEntries": []
}
```

자세한 내용은 IoT 이벤트 참조BatchPutMessage의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조BatchPutMessage의 섹션을 참조하세요. AWS CLI

### `batch-update-detector`

다음 코드 예시에서는 `batch-update-detector`을 사용하는 방법을 보여 줍니다.

AWS CLI

감지기를 업데이트하려면(인스턴스)

다음 `batch-update-detector` 예제에서는 지정된 감지기 모델의 하나 이상의 감지기(인스턴스)의 상태, 변수 값 및 타이머 설정을 업데이트합니다.

```
aws iotevents-data batch-update-detector \
    --cli-input-json file://budFulton-A32.json
```

`budFulton-A32.json`의 콘텐츠:

```
{
    "detectors": [
        {
```

```
            "messageId": "00001",
            "detectorModelName": "motorDetectorModel",
            "keyValue": "Fulton-A32",
            "state": {
                "stateName": "Normal",
                "variables": [
                    {
                        "name": "pressureThresholdBreached",
                        "value": "0"
                    }
                ],
                "timers": [
                ]
            }
        }
    ]
}
```

출력:

```
{
    "batchUpdateDetectorErrorEntries": []
}
```

자세한 내용은 IoT 이벤트 참조BatchUpdateDetector의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조BatchUpdateDetector의 섹션을 참조하세요. AWS CLI

### `create-detector-model`

다음 코드 예시에서는 `create-detector-model`을 사용하는 방법을 보여 줍니다.

AWS CLI

감지기 모델을 생성하려면

다음 `create-detector-model` 예제에서는 파라미터 파일에 지정된 구성을 사용하여 감지기 모델을 생성합니다.

```
aws iotevents create-detector-model  \
    --cli-input-json file://motorDetectorModel.json
```

motorDetectorModel.json의 콘텐츠:

```
{
    "detectorModelName": "motorDetectorModel",
    "detectorModelDefinition": {
        "states": [
            {
                "stateName": "Normal",
                "onEnter": {
                    "events": [
                        {
                            "eventName": "init",
                            "condition": "true",
                            "actions": [
                                {
                                    "setVariable": {
                                        "variableName": "pressureThresholdBreached",
                                        "value": "0"
                                    }
                                }
                            ]
                        }
                    ]
                },
                "onInput": {
                    "transitionEvents": [
                        {
                            "eventName": "Overpressurized",
                            "condition": "$input.PressureInput.sensorData.pressure
 &gt; 70",
                            "actions": [
                                {
                                    "setVariable": {
                                        "variableName": "pressureThresholdBreached",
                                        "value":
 "$variable.pressureThresholdBreached + 3"
                                    }
                                }
                            ],
                            "nextState": "Dangerous"
                        }
                    ]
                }
            },
```

```
            {
                "stateName": "Dangerous",
                "onEnter": {
                    "events": [
                        {
                            "eventName": "Pressure Threshold Breached",
                            "condition": "$variable.pressureThresholdBreached &gt;
 1",
                            "actions": [
                                {
                                    "sns": {
                                        "targetArn": "arn:aws:sns:us-
east-1:123456789012:underPressureAction"
                                    }
                                }
                            ]
                        }
                    ]
                },
                "onInput": {
                    "events": [
                        {
                            "eventName": "Overpressurized",
                            "condition": "$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"
}
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 IoT 이벤트 참조CreateDetectorModel의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조CreateDetectorModel의 섹션을 참조하세요. AWS CLI

### `create-input`

다음 코드 예시에서는 `create-input`을 사용하는 방법을 보여 줍니다.

AWS CLI

입력을 생성하려면

다음 `create-input` 예제에서는 입력을 생성합니다.

```
aws iotevents create-input  \
    --cli-input-json file://pressureInput.json
```

`pressureInput.json`의 콘텐츠:

```
{
    "inputName": "PressureInput",
    "inputDescription": "Pressure readings from a motor",
    "inputDefinition": {
        "attributes": [
            { "jsonPath": "sensorData.pressure" },
            { "jsonPath": "motorid" }
        ]
    }
}
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 IoT 이벤트 참조CreateInput의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조CreateInput의 섹션을 참조하세요. AWS CLI

### `delete-detector-model`

다음 코드 예시에서는 `delete-detector-model`을 사용하는 방법을 보여 줍니다.

AWS CLI

감지기 모델을 삭제하려면

다음 `delete-detector-model` 예제에서는 지정된 감지기 모델을 삭제합니다. 감지기 모델의 모든 활성 인스턴스도 삭제됩니다.

```
aws iotevents delete-detector-model \
    --detector-model-name motorDetectorModel
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IoT 이벤트 참조DeleteDetectorModel의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조DeleteDetectorModel의 섹션을 참조하세요. AWS CLI

### `delete-input`

다음 코드 예시에서는 `delete-input`을 사용하는 방법을 보여 줍니다.

AWS CLI

입력을 삭제하려면

다음 delete-input 예제에서는 지정된 입력을 삭제합니다.

```
aws iotevents delete-input \
    --input-name PressureInput
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IoT 이벤트 참조DeleteInput의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조DeleteInput의 섹션을 참조하세요. AWS CLI

### describe-detector-model

다음 코드 예시에서는 describe-detector-model을 사용하는 방법을 보여 줍니다.

AWS CLI

감지기 모델에 대한 정보를 가져오려면

다음 describe-detector-model 예제에서는 지정된 감지기 모델에 대한 세부 정보를 표시합니다. version 파라미터가 지정되지 않았으므로 최신 버전에 대한 정보가 반환됩니다.

```
aws iotevents describe-detector-model \
    --detector-model-name motorDetectorModel
```

출력:

```
{
    "detectorModel": {
        "detectorModelConfiguration": {
            "status": "ACTIVE",
            "lastUpdateTime": 1560796816.077,
            "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
            "creationTime": 1560796816.077,
            "detectorModelArn": "arn:aws:iotevents:us-
west-2:123456789012:detectorModel/motorDetectorModel",
            "key": "motorid",
            "detectorModelName": "motorDetectorModel",
            "detectorModelVersion": "1"
        },
        "detectorModelDefinition": {
            "states": [
                {
```

```
                    "onInput": {
                        "transitionEvents": [
                            {
                                "eventName": "Overpressurized",
                                "actions": [
                                    {
                                        "setVariable": {
                                            "variableName":
"pressureThresholdBreached",
                                            "value":
"$variable.pressureThresholdBreached + 3"
                                        }
                                    }
                                ],
                                "condition":
"$input.PressureInput.sensorData.pressure > 70",
                                "nextState": "Dangerous"
                            }
                        ],
                        "events": []
                    },
                    "stateName": "Normal",
                    "onEnter": {
                        "events": [
                            {
                                "eventName": "init",
                                "actions": [
                                    {
                                        "setVariable": {
                                            "variableName":
"pressureThresholdBreached",
                                            "value": "0"
                                        }
                                    }
                                ],
                                "condition": "true"
                            }
                        ]
                    },
                    "onExit": {
                        "events": []
                    }
                },
                {
```

```
                    "onInput": {
                        "transitionEvents": [
                            {
                                "eventName": "BackToNormal",
                                "actions": [],
                                "condition":
"$input.PressureInput.sensorData.pressure <= 70 &&
$variable.pressureThresholdBreached <= 1",
                                "nextState": "Normal"
                            }
                        ],
                        "events": [
                            {
                                "eventName": "Overpressurized",
                                "actions": [
                                    {
                                        "setVariable": {
                                            "variableName":
"pressureThresholdBreached",
                                            "value": "3"
                                        }
                                    }
                                ],
                                "condition":
"$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": [
                                    {
                                        "sns": {
                                            "targetArn": "arn:aws:sns:us-
east-1:123456789012:pressureClearedAction"
                                        }
                                    }
                                ],
                                "condition": "true"
                            }
                        ]
                    }
                }
            ],
            "initialStateName": "Normal"
        }
    }
}
```

자세한 내용은 IoT 이벤트 참조DescribeDetectorModel의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조DescribeDetectorModel의 섹션을 참조하세요. AWS CLI

### describe-detector

다음 코드 예시에서는 describe-detector을 사용하는 방법을 보여 줍니다.

AWS CLI

감지기(인스턴스)에 대한 정보를 가져옵니다.

다음 describe-detector 예제에서는 지정된 감지기(인스턴스)에 대한 세부 정보를 표시합니다.

```
aws iotevents-data describe-detector \
    --detector-model-name motorDetectorModel \
    --key-value "Fulton-A32"
```

출력:

```
{
    "detector": {
        "lastUpdateTime": 1560797852.776,
        "creationTime": 1560797852.775,
        "state": {
            "variables": [
                {
                    "name": "pressureThresholdBreached",
                    "value": "3"
                }
            ],
            "stateName": "Dangerous",
            "timers": []
        },
        "keyValue": "Fulton-A32",
        "detectorModelName": "motorDetectorModel",
        "detectorModelVersion": "1"
    }
}
```

자세한 내용은 IoT 이벤트 참조DescribeDetector의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조DescribeDetector의 섹션을 참조하세요. AWS CLI

### describe-input

다음 코드 예시에서는 describe-input을 사용하는 방법을 보여 줍니다.

AWS CLI

입력에 대한 정보를 가져오려면

다음 `describe-input` 예제에서는 지정된 입력에 대한 세부 정보를 표시합니다.

```
aws iotevents describe-input \
    --input-name PressureInput
```

출력:

```
{
    "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"
                }
            ]
        }
    }
}
```

자세한 내용은 IoT 이벤트 참조DescribeInput의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조DescribeInput의 섹션을 참조하세요. AWS CLI

### `describe-logging-options`

다음 코드 예시에서는 `describe-logging-options`을 사용하는 방법을 보여 줍니다.

AWS CLI

로깅 설정에 대한 정보를 가져오려면

다음 `describe-logging-options` 예제에서는 AWS IoT Events 로깅 옵션의 현재 설정을 검색합니다.

```
aws iotevents describe-logging-options
```

출력:

```
{
    "loggingOptions": {
        "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
        "enabled": false,
        "level": "ERROR"
    }
}
```

자세한 내용은 IoT 이벤트 참조DescribeLoggingOptions의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조DescribeLoggingOptions의 섹션을 참조하세요. AWS CLI

### `list-detector-model-versions`

다음 코드 예시에서는 `list-detector-model-versions`을 사용하는 방법을 보여 줍니다.

AWS CLI

감지기 모델 버전에 대한 정보를 가져오려면

다음 `list-detector-model-versions` 예제에서는 감지기 모델의 모든 버전을 나열합니다. 각 감지기 모델 버전과 연결된 메타데이터만 반환됩니다.

```
aws iotevents list-detector-model-versions \
    --detector-model-name motorDetectorModel
```

출력:

```
{
```

```
    "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"
        }
    ]
}
```

자세한 내용은 IoT 이벤트 참조ListDetectorModelVersions의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조ListDetectorModelVersions의 섹션을 참조하세요. AWS CLI

### `list-detector-models`

다음 코드 예시에서는 `list-detector-models`을 사용하는 방법을 보여 줍니다.

AWS CLI

감지기 모델 목록을 가져오려면

다음 `list-detector-models` 예제에서는 생성한 감지기 모델을 나열합니다. 각 감지기 모델과 연결된 메타데이터만 반환됩니다.

```
aws iotevents list-detector-models
```

출력:

```
{
    "detectorModelSummaries": [
        {
            "detectorModelName": "motorDetectorModel",
            "creationTime": 1552072424.212
            "detectorModelDescription": "Detect overpressure in a motor."
        }
    ]
}
```

자세한 내용은 IoT 이벤트 참조ListDetectorModels의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조ListDetectorModels의 섹션을 참조하세요. AWS CLI

### `list-detectors`

다음 코드 예시에서는 `list-detectors`을 사용하는 방법을 보여 줍니다.

AWS CLI

감지기 모델의 감지기 목록을 가져오려면

다음 `list-detectors` 예제에서는 계정의 감지기(감지기 모델의 인스턴스)를 나열합니다.

```
aws iotevents-data list-detectors \
    --detector-model-name motorDetectorModel
```

출력:

```
{
    "detectorSummaries": [
        {
            "lastUpdateTime": 1558129925.2,
            "creationTime": 1552073155.527,
            "state": {
                "stateName": "Normal"
            },
            "keyValue": "Fulton-A32",
            "detectorModelName": "motorDetectorModel",
            "detectorModelVersion": "1"
        }
    ]
}
```

자세한 내용은 IoT 이벤트 참조ListDetectors의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조ListDetectors의 섹션을 참조하세요. AWS CLI

### `list-inputs`

다음 코드 예시에서는 `list-inputs`을 사용하는 방법을 보여 줍니다.

AWS CLI

입력을 나열하려면

다음 `list-inputs` 예제에서는 계정에서 생성한 입력을 나열합니다.

```
aws iotevents list-inputs
```

이 명령은 출력을 생성하지 않습니다. 출력:

```
{
    {
        "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"
    }
}
```

자세한 내용은 IoT 이벤트 참조ListInputs의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조ListInputs의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 할당된 태그를 나열합니다.

다음 `list-tags-for-resource` 예제에서는 리소스에 할당한 태그 키 이름과 값을 나열합니다.

```
aws iotevents list-tags-for-resource \
    --resource-arn "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput"
```

출력:

```
{
    "tags": [
```

```
        {
            "value": "motor",
            "key": "deviceType"
        }
    ]
}
```

자세한 내용은 IoT 이벤트 참조ListTagsForResource의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### put-logging-options

다음 코드 예시에서는 put-logging-options을 사용하는 방법을 보여 줍니다.

AWS CLI

로깅 옵션을 설정하려면

다음 put-logging-options 예제에서는 AWS IoT Events 로깅 옵션을 설정하거나 업데이트합니다. 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 필드의 값을 업데이트하는 경우(예: 잘못된 정책을 수정하는 경우) 변경 사항이 적용되려면 최대 5분이 걸릴 수 있습니다.

```
aws iotevents put-logging-options \
    --cli-input-json file://logging-options.json
```

logging-options.json의 콘텐츠:

```
{
    "loggingOptions": {
        "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
        "level": "DEBUG",
        "enabled": true,
        "detectorDebugOptions": [
            {
                "detectorModelName": "motorDetectorModel",
                "keyValue": "Fulton-A32"
            }
        ]
    }
```

```
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IoT 이벤트 참조PutLoggingOptions의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조PutLoggingOptions의 섹션을 참조하세요. AWS CLI

### `tag-resource`

다음 코드 예시에서는 `tag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 태그를 추가하려면

다음 `tag-resource` 예제에서는 지정된 리소스에 연결된 태그를 추가하거나 수정합니다(키가 `deviceType` 이미 있는 경우).

```
aws iotevents tag-resource \
    --cli-input-json file://pressureInput.tag.json
```

`pressureInput.tag.json`의 콘텐츠:

```
{
    "resourceArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",
    "tags": [
        {
            "key": "deviceType",
            "value": "motor"
        }
    ]
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IoT 이벤트 참조TagResource의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### `untag-resource`

다음 코드 예시에서는 `untag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에서 태그를 제거하려면

다음 `untag-resource` 예제에서는 지정된 리소스에서 지정된 키 이름을 가진 태그를 제거합니다.

```
aws iotevents untag-resource \
    --resource-arn arn:aws:iotevents:us-west-2:123456789012:input/PressureInput \
    --tagkeys deviceType
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IoT 이벤트 참조UntagResource의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-detector-model

다음 코드 예시에서는 `update-detector-model`을 사용하는 방법을 보여 줍니다.

AWS CLI

감지기 모델을 업데이트하려면

다음 `update-detector-model` 예제에서는 지정된 감지기 모델을 업데이트합니다. 이전 버전에서 생성된 탐지기(인스턴스)는 삭제된 다음 새 입력이 도착하면 다시 생성됩니다.

```
aws iotevents update-detector-model \
    --cli-input-json file://motorDetectorModel.update.json
```

`motorDetectorModel.update.json`의 콘텐츠:

```
{
    "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"
}
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 IoT 이벤트 참조UpdateDetectorModel의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조UpdateDetectorModel의 섹션을 참조하세요. AWS CLI

### update-input

다음 코드 예시에서는 update-input을 사용하는 방법을 보여 줍니다.

AWS CLI

입력을 업데이트하려면

다음 update-input 예제에서는 지정된 입력을 새 설명 및 정의로 업데이트합니다.

```
aws iotevents update-input \
    --cli-input-json file://pressureInput.json
```

pressureInput.json의 콘텐츠:

```
{
    "inputName": "PressureInput",
    "inputDescription": "Pressure readings from a motor",
    "inputDefinition": {
        "attributes": [
            { "jsonPath": "sensorData.pressure" },
            { "jsonPath": "motorid" }
        ]
    }
}
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 IoT 이벤트 참조UpdateInput의 섹션을 참조하세요. AWS IoT API

- 자세한 API 내용은 명령 참조UpdateInput의 섹션을 참조하세요. AWS CLI

# AWS IoT Events-Data 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS IoT Events-Data.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `batch-put-message`

다음 코드 예시에서는 `batch-put-message`을 사용하는 방법을 보여 줍니다.

AWS CLI

메시지(입력)를 AWS IoT Events로 보내려면

다음 `batch-put-message` 예제에서는 AWS IoT Events 시스템에 메시지 세트를 전송합니다. 각 메시지 페이로드는 지정한 입력(`inputName`)으로 변환되고 해당 입력을 모니터링하는 모든 탐지기로 수집됩니다. 여러 메시지가 전송되는 경우 메시지가 처리되는 순서가 보장되지 않습니다. 주문을 보장하려면 메시지를 한 번에 하나씩 보내고 응답이 성공할 때까지 기다려야 합니다.

```
aws iotevents-data batch-put-message \
    --cli-binary-format raw-in-base64-out \
    --cli-input-json file://highPressureMessage.json
```

`highPressureMessage.json`의 콘텐츠:

```
{
    "messages": [
        {
            "messageId": "00001",
            "inputName": "PressureInput",
```

```
            "payload": "{\"motorid\": \"Fulton-A32\", \"sensorData\": {\"pressure\":
 80, \"temperature\": 39} }"
        }
    ]
}
```

출력:

```
{
    "BatchPutMessageErrorEntries": []
}
```

자세한 내용은 AWS IoT Events 개발자 안내서*BatchPutMessage의 섹션을 참조하세요.

- 자세한 API 내용은 명령 참조BatchPutMessage의 섹션을 참조하세요. AWS CLI

### batch-update-detector

다음 코드 예시에서는 batch-update-detector을 사용하는 방법을 보여 줍니다.

AWS CLI

감지기를 업데이트하려면(인스턴스)

다음 batch-update-detector 예제에서는 지정된 감지기 모델의 하나 이상의 감지기(인스턴스)의 상태, 변수 값 및 타이머 설정을 업데이트합니다.

```
aws iotevents-data batch-update-detector \
    --cli-input-json file://budFulton-A32.json
```

budFulton-A32.json의 콘텐츠:

```
{
    "detectors": [
        {
            "messageId": "00001",
            "detectorModelName": "motorDetectorModel",
            "keyValue": "Fulton-A32",
            "state": {
                "stateName": "Normal",
                "variables": [
                    {
```

```
                        "name": "pressureThresholdBreached",
                        "value": "0"
                    }
                ],
                "timers": [
                ]
            }
        }
    ]
}
```

출력:

```
{
    "batchUpdateDetectorErrorEntries": []
}
```

자세한 내용은 AWS IoT Events 개발자 안내서*BatchUpdateDetector의 섹션을 참조하세요.

- 자세한 API 내용은 명령 참조BatchUpdateDetector의 섹션을 참조하세요. AWS CLI

### `create-detector-model`

다음 코드 예시에서는 `create-detector-model`을 사용하는 방법을 보여 줍니다.

AWS CLI

감지기 모델을 생성하려면

다음 `create-detector-model` 예제에서는 감지기 모델을 생성합니다.

```
aws iotevents create-detector-model \
    --cli-input-json file://motorDetectorModel.json
```

`motorDetectorModel.json`의 콘텐츠:

```
{
    "detectorModelName": "motorDetectorModel",
    "detectorModelDefinition": {
        "states": [
            {
                "stateName": "Normal",
                "onEnter": {
```

```
                    "events": [
                        {
                            "eventName": "init",
                            "condition": "true",
                            "actions": [
                                {
                                    "setVariable": {
                                        "variableName": "pressureThresholdBreached",
                                        "value": "0"
                                    }
                                }
                            ]
                        }
                    ]
                },
                "onInput": {
                    "transitionEvents": [
                        {
                            "eventName": "Overpressurized",
                            "condition": "$input.PressureInput.sensorData.pressure
&gt; 70",
                            "actions": [
                                {
                                    "setVariable": {
                                        "variableName": "pressureThresholdBreached",
                                        "value":
"$variable.pressureThresholdBreached + 3"
                                    }
                                }
                            ],
                            "nextState": "Dangerous"
                        }
                    ]
                }
            },
            {
                "stateName": "Dangerous",
                "onEnter": {
                    "events": [
                        {
                            "eventName": "Pressure Threshold Breached",
                            "condition": "$variable.pressureThresholdBreached &gt;
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"
}
```

출력:

```
{
    "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"
```

```
    }
}
```

자세한 내용은 AWS IoT Events 개발자 안내서*CreateDetectorModel의 섹션을 참조하세요.

- 자세한 API 내용은 명령 참조CreateDetectorModel의 섹션을 참조하세요. AWS CLI

### `create-input`

다음 코드 예시에서는 `create-input`을 사용하는 방법을 보여 줍니다.

AWS CLI

입력을 생성하려면

다음 `create-input` 예제에서는 입력을 생성합니다.

```
aws iotevents create-input \
    --cli-input-json file://pressureInput.json
```

`pressureInput.json`의 콘텐츠:

```
{
    "inputName": "PressureInput",
    "inputDescription": "Pressure readings from a motor",
    "inputDefinition": {
        "attributes": [
                { "jsonPath": "sensorData.pressure" },
                { "jsonPath": "motorid" }
            ]
    }
}
```

출력:

```
{
    "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"
```

```
    }
}
```

자세한 내용은 AWS IoT Events 개발자 안내서*CreateInput의 섹션을 참조하세요.

- 자세한 API 내용은 명령 참조CreateInput의 섹션을 참조하세요. AWS CLI

### delete-detector-model

다음 코드 예시에서는 delete-detector-model을 사용하는 방법을 보여 줍니다.

AWS CLI

감지기 모델을 삭제하려면

다음 delete-detector-model 예제에서는 감지기 모델을 삭제합니다. 감지기 모델의 모든 활성 인스턴스도 삭제됩니다.

```
aws iotevents delete-detector-model \
    --detector-model-name motorDetectorModel*
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT Events 개발자 안내서*DeleteDetectorModel의 섹션을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteDetectorModel의 섹션을 참조하세요. AWS CLI

### delete-input

다음 코드 예시에서는 delete-input을 사용하는 방법을 보여 줍니다.

AWS CLI

입력을 삭제하려면

다음 delete-input 예제에서는 입력을 삭제합니다.

```
aws iotevents delete-input \
    --input-name PressureInput
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT Events 개발자 안내서*DeleteInput의 섹션을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteInput의 섹션을 참조하세요. AWS CLI

### `describe-detector-model`

다음 코드 예시에서는 `describe-detector-model`을 사용하는 방법을 보여 줍니다.

AWS CLI

감지기 모델에 대한 정보를 가져오려면

다음 `describe-detector-model` 예제에서는 감지기 모델을 설명합니다. `version` 파라미터를 지정하지 않으면 명령은 최신 버전에 대한 정보를 반환합니다.

```
aws iotevents describe-detector-model \
    --detector-model-name motorDetectorModel
```

출력:

```
{
    "detectorModel": {
        "detectorModelConfiguration": {
            "status": "ACTIVE",
            "lastUpdateTime": 1560796816.077,
            "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
            "creationTime": 1560796816.077,
            "detectorModelArn": "arn:aws:iotevents:us-
west-2:123456789012:detectorModel/motorDetectorModel",
            "key": "motorid",
            "detectorModelName": "motorDetectorModel",
            "detectorModelVersion": "1"
        },
        "detectorModelDefinition": {
            "states": [
                {
                    "onInput": {
                        "transitionEvents": [
                            {
                                "eventName": "Overpressurized",
                                "actions": [
                                    {
                                        "setVariable": {
                                            "variableName":
 "pressureThresholdBreached",
```

```
                                            "value":
"$variable.pressureThresholdBreached + 3"
                                        }
                                    }
                                ],
                                "condition":
"$input.PressureInput.sensorData.pressure > 70",
                                "nextState": "Dangerous"
                            }
                        ],
                        "events": []
                    },
                    "stateName": "Normal",
                    "onEnter": {
                        "events": [
                            {
                                "eventName": "init",
                                "actions": [
                                    {
                                        "setVariable": {
                                            "variableName":
"pressureThresholdBreached",
                                            "value": "0"
                                        }
                                    }
                                ],
                                "condition": "true"
                            }
                        ]
                    },
                    "onExit": {
                        "events": []
                    }
                },
                {
                    "onInput": {
                        "transitionEvents": [
                            {
                                "eventName": "BackToNormal",
                                "actions": [],
                                "condition":
"$input.PressureInput.sensorData.pressure <= 70 &&
$variable.pressureThresholdBreached <= 1",
                                "nextState": "Normal"
```

```
                            }
                        ],
                        "events": [
                            {
                                "eventName": "Overpressurized",
                                "actions": [
                                    {
                                        "setVariable": {
                                            "variableName":
 "pressureThresholdBreached",
                                            "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": [
                                {
                                    "sns": {
                                        "targetArn": "arn:aws:sns:us-
east-1:123456789012:pressureClearedAction"
                                    }
                                }
                            ],
                            "condition": "true"
                        }
                    ]
                }
            }
        ],
        "initialStateName": "Normal"
    }
}
}
```

자세한 내용은 AWS IoT Events 개발자 안내서*DescribeDetectorModel의 섹션을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeDetectorModel의 섹션을 참조하세요. AWS CLI

### `describe-detector`

다음 코드 예시에서는 `describe-detector`을 사용하는 방법을 보여 줍니다.

AWS CLI

감지기에 대한 정보를 가져오려면(인스턴스)

다음 `describe-detector` 예제에서는 지정된 감지기(인스턴스)에 대한 정보를 반환합니다.

```
aws iotevents-data describe-detector \
    --detector-model-name motorDetectorModel \
    --key-value "Fulton-A32"
```

출력:

```
{
    "detector": {
        "lastUpdateTime": 1560797852.776,
        "creationTime": 1560797852.775,
        "state": {
            "variables": [
                {
                    "name": "pressureThresholdBreached",
                    "value": "3"
                }
            ],
            "stateName": "Dangerous",
            "timers": []
        },
        "keyValue": "Fulton-A32",
        "detectorModelName": "motorDetectorModel",
        "detectorModelVersion": "1"
    }
}
```

자세한 내용은 AWS IoT Events 개발자 안내서*DescribeDetector의 섹션을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeDetector의 섹션을 참조하세요. AWS CLI

### `describe-input`

다음 코드 예시에서는 `describe-input`을 사용하는 방법을 보여 줍니다.

AWS CLI

입력에 대한 정보를 가져오려면

다음 `describe-input` 예제에서는 입력의 세부 정보를 검색합니다.

```
aws iotevents describe-input \
    --input-name PressureInput
```

출력:

```
{
    "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"
                }
            ]
        }
    }
}
```

자세한 내용은 AWS IoT Events 개발자 안내서*DescribeInput의 섹션을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeInput의 섹션을 참조하세요. AWS CLI

### describe-logging-options

다음 코드 예시에서는 describe-logging-options을 사용하는 방법을 보여 줍니다.

AWS CLI

로깅 설정에 대한 정보를 가져오려면

다음 describe-logging-options 예제에서는 현재 AWS IoT Events 로깅 옵션을 검색합니다.

```
aws iotevents describe-logging-options
```

출력:

```
{
    "loggingOptions": {
        "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
        "enabled": false,
        "level": "ERROR"
    }
}
```

자세한 내용은 AWS IoT Events 개발자 안내서*DescribeLoggingOptions의 섹션을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeLoggingOptions의 섹션을 참조하세요. AWS CLI

### `list-detector-model-versions`

다음 코드 예시에서는 `list-detector-model-versions`을 사용하는 방법을 보여 줍니다.

AWS CLI

감지기 모델 버전에 대한 정보를 가져오려면

다음 `list-detector-model-versions` 예제에서는 감지기 모델의 모든 버전을 나열합니다. 각 감지기 모델 버전과 연결된 메타데이터만 반환됩니다.

```
aws iotevents list-detector-model-versions \
    --detector-model-name motorDetectorModel
```

출력:

```
{
    "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"
        }
    ]
```

```
}
```

자세한 내용은 AWS IoT Events 개발자 안내서*ListDetectorModelVersions의 섹션을 참조하세요.

- 자세한 API 내용은 명령 참조ListDetectorModelVersions의 섹션을 참조하세요. AWS CLI

### `list-detector-models`

다음 코드 예시에서는 `list-detector-models`을 사용하는 방법을 보여 줍니다.

AWS CLI

감지기 모델 목록을 가져오려면

다음 `list-detector-models` 예제에서는 생성한 감지기 모델을 나열합니다. 각 감지기 모델과 연결된 메타데이터만 반환됩니다.

```
aws iotevents list-detector-models
```

출력:

```
{
    "detectorModelSummaries": [
        {
            "detectorModelName": "motorDetectorModel",
            "creationTime": 1552072424.212
            "detectorModelDescription": "Detect overpressure in a motor."
        }
    ]
}
```

자세한 내용은 AWS IoT Events 개발자 안내서*ListDetectorModels의 섹션을 참조하세요.

- 자세한 API 내용은 명령 참조ListDetectorModels의 섹션을 참조하세요. AWS CLI

### `list-detectors`

다음 코드 예시에서는 `list-detectors`을 사용하는 방법을 보여 줍니다.

AWS CLI

감지기 모델의 감지기 목록을 가져오려면

다음 `list-detectors` 예제에서는 감지기(감지기 모델의 인스턴스)를 나열합니다.

```
aws iotevents-data list-detectors \
    --detector-model-name motorDetectorModel
```

출력:

```
{
    "detectorSummaries": [
        {
            "lastUpdateTime": 1558129925.2,
            "creationTime": 1552073155.527,
            "state": {
                "stateName": "Normal"
            },
            "keyValue": "Fulton-A32",
            "detectorModelName": "motorDetectorModel",
            "detectorModelVersion": "1"
        }
    ]
}
```

자세한 내용은 AWS IoT Events 개발자 안내서*ListDetectors의 섹션을 참조하세요.

- 자세한 API 내용은 명령 참조ListDetectors의 섹션을 참조하세요. AWS CLI

### `list-inputs`

다음 코드 예시에서는 `list-inputs`을 사용하는 방법을 보여 줍니다.

AWS CLI

입력을 나열하려면

다음 `list-inputs` 예제에서는 생성한 입력을 나열합니다.

```
aws iotevents list-inputs
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS IoT Events 개발자 안내서*ListInputs의 섹션을 참조하세요.

- 자세한 API 내용은 명령 참조ListInputs의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 할당된 태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 리소스에 할당한 태그(메타데이터)를 나열합니다.

```
aws iotevents list-tags-for-resource \
    --resource-arn "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput"
```

출력:

```
{
    "tags": [
        {
            "value": "motor",
            "key": "deviceType"
        }
    ]
}
```

자세한 내용은 AWS IoT Events 개발자 안내서*ListTagsForResource의 섹션을 참조하세요.

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### `put-logging-options`

다음 코드 예시에서는 `put-logging-options`을 사용하는 방법을 보여 줍니다.

AWS CLI

로깅 옵션을 설정하려면

다음 `list-tags-for-resource` 예제에서는 AWS IoT Events 로깅 옵션을 설정하거나 업데이트합니다. `loggingOptions` 필드 값을 업데이트한 경우 변경 사항이 적용되기까지 최대 1분이 소요될 수 있습니다. 또한 `roleArn` 필드에 지정한 역할에 연결된 정책을 변경하는 경우(예: 잘못된 정책을 수정하는 경우) 해당 변경 사항이 적용되려면 최대 5분이 걸립니다.

```
aws iotevents put-logging-options \
    --cli-input-json file://logging-options.json
```

`logging-options.json`의 콘텐츠:

```
{
    "loggingOptions": {
        "roleArn": "arn:aws:iam::123456789012:role/IoTEventsRole",
        "level": "DEBUG",
        "enabled": true,
        "detectorDebugOptions": [
            {
                "detectorModelName": "motorDetectorModel",
                "keyValue": "Fulton-A32"
            }
        ]
    }
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT Events 개발자 안내서*PutLoggingOptions의 섹션을 참조하세요.

- 자세한 API 내용은 명령 참조PutLoggingOptions의 섹션을 참조하세요. AWS CLI

## `tag-resource`

다음 코드 예시에서는 `tag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 태그를 추가하려면

다음 tag-resource 예제에서는 지정된 리소스의 태그를 추가하거나 수정합니다. 태그는 리소스를 관리하는 데 사용할 수 있는 메타데이터입니다.

```
aws iotevents tag-resource \
    --cli-input-json file://pressureInput.tag.json
```

pressureInput.tag.json의 콘텐츠:

```
{
    "resourceArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",
    "tags": [
        {
            "key": "deviceType",
            "value": "motor"
        }
    ]
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT Events 개발자 안내서*TagResource의 섹션을 참조하세요.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### untag-resource

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에서 태그를 제거하려면

다음 untag-resource 예제에서는 리소스에서 지정된 태그를 제거합니다.

```
aws iotevents untag-resource \
    --cli-input-json file://pressureInput.untag.json
```

pressureInput.untag.json의 콘텐츠:

```
{
    "resourceArn": "arn:aws:iotevents:us-west-2:123456789012:input/PressureInput",
    "tagKeys": [
```

```
            "deviceType"
    ]
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT Events 개발자 안내서*UntagResource의 섹션을 참조하세요.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-detector-model

다음 코드 예시에서는 update-detector-model을 사용하는 방법을 보여 줍니다.

AWS CLI

감지기 모델을 업데이트하려면

다음 update-detector-model 예제에서는 감지기 모델을 업데이트합니다. 이전 버전에서 생성된 탐지기(인스턴스)는 삭제된 다음 새 입력이 도착하면 다시 생성됩니다.

```
aws iotevents update-detector-model \
    --cli-input-json file://motorDetectorModel.update.json
```

motorDetectorModel.update.json의 내용:

```
{
  "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"
}
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS IoT Events 개발자 안내서*UpdateDetectorModel의 섹션을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateDetectorModel의 섹션을 참조하세요. AWS CLI

### update-input

다음 코드 예시에서는 update-input을 사용하는 방법을 보여 줍니다.

AWS CLI

입력을 업데이트하려면

다음 update-input 예제에서는 입력을 업데이트합니다.

```
aws iotevents update-input \
```

```
    --cli-input-json file://pressureInput.json
```

pressureInput.json의 콘텐츠:

```
{
    "inputName": "PressureInput",
    "inputDescription": "Pressure readings from a motor",
    "inputDefinition": {
        "attributes": [
            { "jsonPath": "sensorData.pressure" },
            { "jsonPath": "motorid" }
        ]
    }
}
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS IoT Events 개발자 안내서*UpdateInput의 섹션을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateInput의 섹션을 참조하세요. AWS CLI

## AWS IoT Greengrass 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS IoT Greengrass.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### `associate-role-to-group`

다음 코드 예시에서는 `associate-role-to-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

역할을 Greengrass 그룹과 연결하려면

다음 `associate-role-to-group` 예제에서는 지정된 IAM 역할을 Greengrass 그룹과 연결합니다. 그룹 역할은 로컬 Lambda 함수 및 커넥터에서 AWS 서비스에 액세스하는 데 사용됩니다. 예를 들어 그룹 역할은 CloudWatch 로그 통합에 필요한 권한을 부여할 수 있습니다.

```
aws greengrass associate-role-to-group \
    --group-id 2494ee3f-7f8a-4e92-a78b-d205f808b84b \
    --role-arn arn:aws:iam::123456789012:role/GG-Group-Role
```

출력:

```
{
    "AssociatedAt": "2019-09-10T20:03:30Z"
}
```

자세한 내용은 AWS IoT Greengrass 개발자 안내서의 그룹 역할 구성을 참조하세요.

- 자세한 API 내용은 명령 참조AssociateRoleToGroup의 섹션을 참조하세요. AWS CLI

### `associate-service-role-to-account`

다음 코드 예시에서는 `associate-service-role-to-account`을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스 역할을 AWS 계정과 연결하려면

다음 associate-service-role-to-account 예제는 에서 지정한 IAM 서비스 역할을 AWS 계정의 AWS IoT GreengrassARN와 연결합니다. 이전에 에서 서비스 역할을 생성했어야 하며IAM, AWS IoT Greengrass가 이 역할을 수임할 수 있도록 정책 문서를 이와 연결해야 합니다.

```
aws greengrass associate-service-role-to-account \
    --role-arn "arn:aws:iam::123456789012:role/service-role/Greengrass_ServiceRole"
```

출력:

```
{
    "AssociatedAt": "2019-06-25T18:12:45Z"
}
```

자세한 내용은 AWS IoT Greengrass 개발자 안내서의 Greengrass 서비스 역할을 참조하세요.

- 자세한 API 내용은 명령 참조AssociateServiceRoleToAccount의 섹션을 참조하세요. AWS CLI

### create-connector-definition-version

다음 코드 예시에서는 create-connector-definition-version을 사용하는 방법을 보여 줍니다.

AWS CLI

커넥터 정의 버전을 생성하려면

다음 create-connector-definition-version 예제에서는 커넥터 정의 버전을 생성하고 이를 지정된 커넥터 정의와 연결합니다. 버전의 모든 커넥터는 파라미터 값을 정의합니다.

```
aws greengrass create-connector-definition-version \
    --connector-definition-id "55d0052b-0d7d-44d6-b56f-21867215e118" \
    --connectors "[{\"Id\": \"MyTwilioNotificationsConnector\",
 \"ConnectorArn\": \"arn:aws:greengrass:us-west-2::/connectors/
TwilioNotifications/versions/2\", \"Parameters\": {\"TWILIO_ACCOUNT_SID
\": \"AC1a8d4204890840d7fc482aab38090d57\", \"TwilioAuthTokenSecretArn\":
 \"arn:aws:secretsmanager:us-west-2:123456789012:secret:greengrass-TwilioAuthToken-
ntSlp6\", \"TwilioAuthTokenSecretArn-ResourceId\": \"TwilioAuthToken\",
 \"DefaultFromPhoneNumber\": \"4254492999\"}}]"
```

출력:

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
connectors/55d0052b-0d7d-44d6-b56f-21867215e118/versions/33f709a0-c825-49cb-9eea-
dc8964fbd635",
    "CreationTimestamp": "2019-06-24T20:46:30.134Z",
    "Id": "55d0052b-0d7d-44d6-b56f-21867215e118",
    "Version": "33f709a0-c825-49cb-9eea-dc8964fbd635"
}
```

- 자세한 API 내용은 명령 참조CreateConnectorDefinitionVersion의 섹션을 참조하세요. AWS CLI

### `create-connector-definition`

다음 코드 예시에서는 `create-connector-definition`을 사용하는 방법을 보여 줍니다.

AWS CLI

커넥터 정의를 생성하려면

다음 `create-connector-definition` 예제에서는 커넥터 정의와 초기 커넥터 정의 버전을 생성합니다. 초기 버전에는 하나의 커넥터가 포함되어 있습니다. 버전의 모든 커넥터는 파라미터 값을 정의합니다.

```
aws greengrass create-connector-definition \
    --name MySNSConnector \
    --initial-version "{\"Connectors\": [{\"Id\":\"MySNSConnector\",\"ConnectorArn
\":\"arn:aws:greengrass:us-west-2::/connectors/SNS/versions/1\",\"Parameters\":
 {\"DefaultSNSArn\":\"arn:aws:sns:us-west-2:123456789012:GGConnectorTopic\"}}]}"
```

출력:

```
{
    "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"
```

```
}
```

자세한 내용은 IoT Greengrass 개발자 안내서의 Greengrass 커넥터 시작하기(CLI)를 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조CreateConnectorDefinition의 섹션을 참조하세요. AWS CLI

### create-core-definition-version

다음 코드 예시에서는 create-core-definition-version을 사용하는 방법을 보여 줍니다.

AWS CLI

코어 정의 버전을 생성하려면

다음 create-core-definition-version 예제에서는 코어 정의 버전을 생성하고 이를 지정된 코어 정의와 연결합니다. 버전에는 하나의 코어만 포함될 수 있습니다. 코어를 생성하려면 먼저 해당 AWS IoT 사물을 생성하고 프로비저닝해야 합니다. 이 프로세스에는 iot 명령에 CertificateArn 필요한 ThingArn 및 를 반환하는 다음 create-core-definition-version 명령이 포함됩니다.

코어 디바이스에 해당하는 AWS IoT 사물을 생성합니다.

```
aws iot create-thing \
    --thing-name "MyCoreDevice"
```

출력:

```
{
    "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyCoreDevice",
    "thingName": "MyCoreDevice",
    "thingId": "cb419a19-9099-4515-9cec-e9b0e760608a"
}
```

퍼블릭 및 프라이빗 키와 사물에 대한 코어 디바이스 인증서를 생성합니다. 이 예제에서는 create-keys-and-certificate 명령을 사용하며 현재 디렉터리에 대한 쓰기 권한이 필요합니다. 또는 create-certificate-from-csr 명령을 사용할 수 있습니다.

```
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"
```

출력:

```
{
    "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"
}
```

iot 및 greengrass 작업을 허용하는 AWS IoT 정책을 생성합니다. 간소화를 위해 다음 정책은 모든 리소스에 대한 작업을 허용하지만 정책이 더 제한적이어야 합니다.

```
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
\":[\"*\"]}]}"
```

출력:

```
{
    "policyName": "Core_Devices",
    "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/Core_Devices",
    "policyDocument": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\":\"Allow\",\"Action\":[\"iot:Publish\",\"iot:Subscribe\",\"iot:Connect
```

```
\",\"iot:Receive\"],\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":
[\"iot:GetThingShadow\",\"iot:UpdateThingShadow\",\"iot:DeleteThingShadow\"],
\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":[\"greengrass:*\"],\"Resource
\":[\"*\"]}]}",
    "policyVersionId": "1"
}
```

정책을 인증서에 연결합니다.

```
aws iot attach-policy \
    --policy-name "Core_Devices" \
    --target "arn:aws:iot:us-
west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz"
```

이 명령은 출력을 생성하지 않습니다.

인증서에 사물을 연결합니다.

```
aws iot attach-thing-principal \
    --thing-name "MyCoreDevice" \
    --principal "arn:aws:iot:us-
west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz"
```

이 명령은 출력을 생성하지 않습니다.

코어 정의 버전 생성:

```
aws greengrass create-core-definition-version \
    --core-definition-id "582efe12-b05a-409e-9a24-a2ba1bcc4a12" \
    --cores "[{\"Id\":\"MyCoreDevice\",\"ThingArn\":\"arn:aws:iot:us-
west-2:123456789012:thing/MyCoreDevice\",\"CertificateArn\":\"arn:aws:iot:us-
west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz
\",\"SyncShadow\":true}]"
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS IoT Greengrass 개발자 안내서의 IoT Greengrass Core 구성을 참조하세요.
AWS IoT

- 자세한 API 내용은 명령 참조CreateCoreDefinitionVersion의 섹션을 참조하세요. AWS CLI

### `create-core-definition`

다음 코드 예시에서는 `create-core-definition`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 빈 코어 정의를 생성하려면

다음 `create-core-definition` 예제에서는 빈(초기 버전 없음) Greengrass 코어 정의를 생성합니다. 코어를 사용하려면 먼저 `create-core-definition-version` 명령을 사용하여 코어에 대한 다른 파라미터를 제공해야 합니다.

```
aws greengrass create-core-definition \
    --name cliGroup_Core
```

출력:

```
{
    "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"
}
```

예제 2: 초기 버전으로 코어 정의를 생성하려면

다음 `create-core-definition` 예제에서는 초기 코어 정의 버전을 포함하는 코어 정의를 생성합니다. 버전에는 하나의 코어만 포함될 수 있습니다. 코어를 생성하려면 먼저 해당 AWS IoT 사물을 생성하고 프로비저닝해야 합니다. 이 프로세스에는 `iot` 명령에 `CertificateArn` 필요한 `ThingArn` 및 를 반환하는 다음 `create-core-definition` 명령이 포함됩니다.

코어 디바이스에 해당하는 AWS IoT 사물을 생성합니다.

```
aws iot create-thing \
    --thing-name "MyCoreDevice"
```

출력:

```
{
    "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyCoreDevice",
    "thingName": "MyCoreDevice",
    "thingId": "cb419a19-9099-4515-9cec-e9b0e760608a"
}
```

퍼블릭 및 프라이빗 키와 사물에 대한 코어 디바이스 인증서를 생성합니다. 이 예제에서는 `create-keys-and-certificate` 명령을 사용하며 현재 디렉터리에 대한 쓰기 권한이 필요합니다. 또는 `create-certificate-from-csr` 명령을 사용할 수 있습니다.

```
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"
```

출력:

```
{
    "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"
```

```
}
```

iot 및 greengrass 작업을 허용하는 AWS IoT 정책을 생성합니다. 간소화를 위해 다음 정책은 모든 리소스에 대한 작업을 허용하지만 정책이 더 제한적이어야 합니다.

```
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
\":[\"*\"]}]}"
```

출력:

```
{
    "policyName": "Core_Devices",
    "policyArn": "arn:aws:iot:us-west-2:123456789012:policy/Core_Devices",
    "policyDocument": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\":\"Allow\",\"Action\":[\"iot:Publish\",\"iot:Subscribe\",\"iot:Connect
\",\"iot:Receive\"],\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":
[\"iot:GetThingShadow\",\"iot:UpdateThingShadow\",\"iot:DeleteThingShadow\"],
\"Resource\":[\"*\"]},{\"Effect\":\"Allow\",\"Action\":[\"greengrass:*\"],\"Resource
\":[\"*\"]}]}",
    "policyVersionId": "1"
}
```

정책을 인증서에 연결합니다.

```
aws iot attach-policy \
    --policy-name "Core_Devices" \
    --target "arn:aws:iot:us-
west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz"
```

이 명령은 출력을 생성하지 않습니다.

인증서에 사물을 연결합니다.

```
aws iot attach-thing-principal \
    --thing-name "MyCoreDevice" \
```

```
    --principal "arn:aws:iot:us-
west-2:123456789012:cert/123a15ec415668c2349a76170b64ac0878231c1e21ec83c10e92a1EXAMPLExyz"
```

이 명령은 출력을 생성하지 않습니다.

코어 정의를 생성합니다.

```
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}]}"
```

출력:

```
{
    "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/cores/582efe12-b05a-409e-9a24-a2ba1bcc4a12/versions/
cc87b5b3-8f4b-465d-944c-1d6de5dbfcdb",
    "Name": "MyCores",
    "LastUpdatedTimestamp": "2019-09-18T00:11:06.197Z",
    "LatestVersion": "cc87b5b3-8f4b-465d-944c-1d6de5dbfcdb",
    "CreationTimestamp": "2019-09-18T00:11:06.197Z",
    "Id": "582efe12-b05a-409e-9a24-a2ba1bcc4a12",
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/
cores/582efe12-b05a-409e-9a24-a2ba1bcc4a12"
}
```

자세한 내용은 AWS IoT Greengrass 개발자 안내서의 IoT Greengrass Core 구성을 참조하세요.
AWS IoT

- 자세한 API 내용은 명령 참조CreateCoreDefinition의 섹션을 참조하세요. AWS CLI

### `create-deployment`

다음 코드 예시에서는 `create-deployment`을 사용하는 방법을 보여 줍니다.

AWS CLI

Greengrass 그룹의 버전에 대한 배포를 생성하려면

다음 create-deployment 예제에서는 Greengrass 그룹의 지정된 버전을 배포합니다.

```
aws greengrass create-deployment \
    --deployment-type NewDeployment \
    --group-id "ce2e7d01-3240-4c24-b8e6-f6f6e7a9eeca" \
    --group-version-id "dc40c1e9-e8c8-4d28-a84d-a9cad5f599c9"
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS IoT Greengrass 개발자 안내서의 커넥터 시작하기(CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조CreateDeployment의 섹션을 참조하세요. AWS CLI

### create-device-definition-version

다음 코드 예시에서는 create-device-definition-version을 사용하는 방법을 보여 줍니다.

AWS CLI

디바이스 정의 버전을 생성하려면

다음 create-device-definition-version 예제에서는 디바이스 정의 버전을 생성하고 이를 지정된 디바이스 정의와 연결합니다. 버전은 두 개의 디바이스를 정의합니다. Greengrass 디바이스를 생성하려면 먼저 해당 AWS IoT 사물을 생성하고 프로비저닝해야 합니다. 이 프로세스에는 Greengrass iot 명령에 필요한 정보를 얻기 위해 실행해야 하는 다음 명령이 포함됩니다.

디바이스에 해당하는 AWS IoT 사물을 생성합니다.

```
aws iot create-thing \
    --thing-name "InteriorTherm"
```

출력:

```
{
```

```
    "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/InteriorTherm",
    "thingName": "InteriorTherm",
    "thingId": "01d4763c-78a6-46c6-92be-7add080394bf"
}
```

퍼블릭 및 프라이빗 키와 사물에 대한 디바이스 인증서를 생성합니다. 이 예제에서는 `create-keys-and-certificate` 명령을 사용하며 현재 디렉터리에 대한 쓰기 권한이 필요합니다. 또는 다음 `create-certificate-from-csr` 명령을 사용할 수 있습니다.

```
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"
```

출력:

```
{
    "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"
}
```

`iot` 및 `greengrass` 작업을 허용하는 AWS IoT 정책을 생성합니다. 간소화를 위해 다음 정책은 모든 리소스에 대한 작업을 허용하지만 정책이 더 제한적일 수 있습니다.

```
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
\":[\"*\"]}]}"
```

출력:

```
{
    "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"
}
```

정책을 인증서에 연결합니다.

```
aws iot attach-policy \
    --policy-name "GG_Devices" \
    --target "arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92"
```

인증서에 사물 연결

```
aws iot attach-thing-principal \
    --thing-name "InteriorTherm" \
    --principal "arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92"
```

위와 같이 IoT 사물을 생성하고 구성한 후 다음 예제의 처음 두 명령CertificateArn에서 ThingArn 및 를 사용합니다.

```
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}]"
```

출력:

```
{
    "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"
}
```

- 자세한 API 내용은 명령 참조CreateDeviceDefinitionVersion의 섹션을 참조하세요. AWS CLI

### `create-device-definition`

다음 코드 예시에서는 `create-device-definition`을 사용하는 방법을 보여 줍니다.

AWS CLI

디바이스 정의를 생성하려면

다음 `create-device-definition` 예제에서는 초기 디바이스 정의 버전을 포함하는 디바이스 정의를 생성합니다. 초기 버전은 두 디바이스를 정의합니다. Greengrass 디바이스를 생성하려면 먼저 해당 AWS IoT 사물을 생성하고 프로비저닝해야 합니다. 이 프로세스에는 Greengrass `iot` 명령에 필요한 정보를 얻기 위해 실행해야 하는 다음 명령이 포함됩니다.

디바이스에 해당하는 AWS IoT 사물을 생성합니다.

```
aws iot create-thing \
    --thing-name "InteriorTherm"
```

출력:

```
{
    "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/InteriorTherm",
```

```
    "thingName": "InteriorTherm",
    "thingId": "01d4763c-78a6-46c6-92be-7add080394bf"
}
```

퍼블릭 및 프라이빗 키와 사물에 대한 디바이스 인증서를 생성합니다. 이 예제에서는 `create-keys-and-certificate` 명령을 사용하며 현재 디렉터리에 대한 쓰기 권한이 필요합니다. 또는 다음 `create-certificate-from-csr` 명령을 사용할 수 있습니다.

```
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"
```

출력:

```
{
    "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"
}
```

`iot` 및 `greengrass` 작업을 허용하는 AWS IoT 정책을 생성합니다. 간소화를 위해 다음 정책은 모든 리소스에 대한 작업을 허용하지만 정책이 더 제한적일 수 있습니다.

```
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
\":[\"*\"]}]}"
```

출력:

```
{
    "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"
}
```

정책을 인증서에 연결합니다.

```
aws iot attach-policy \
    --policy-name "GG_Devices" \
    --target "arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92"
```

인증서에 사물 연결

```
aws iot attach-thing-principal \
    --thing-name "InteriorTherm" \
    --principal "arn:aws:iot:us-
west-2:123456789012:cert/66a415ec415668c2349a76170b64ac0878231c1e21ec83c10e92a18bd568eb92"
```

위와 같이 IoT 사물을 생성하고 구성한 후 다음 예제의 처음 두 명령CertificateArn에서 ThingArn 및 를 사용합니다.

```
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}]}"
```

출력:

```
{
    "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"
}
```

- 자세한 API 내용은 명령 참조CreateDeviceDefinition의 섹션을 참조하세요. AWS CLI

### `create-function-definition-version`

다음 코드 예시에서는 `create-function-definition-version`을 사용하는 방법을 보여 줍니다.

AWS CLI

함수 정의 버전을 생성하려면

다음 `create-function-definition-version` 예제에서는 지정된 함수 정의의 새 버전을 생성합니다. 이 버전은 ID가 인 단일 함수를 지정하고`Hello-World-function`, 파일 시스템에 대한 액세스를 허용하며, 최대 메모리 크기 및 제한 기간을 지정합니다.

```
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}}]}"
```

출력:

```
{
    "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"
}
```

- 자세한 API 내용은 명령 참조CreateFunctionDefinitionVersion의 섹션을 참조하세요. AWS CLI

### `create-function-definition`

다음 코드 예시에서는 `create-function-definition`을 사용하는 방법을 보여 줍니다.

AWS CLI

Lambda 함수 정의를 생성하려면

다음 `create-function-definition` 예제에서는 Lambda 함수 목록(이 경우 라는 이름의 함수 하나만 목록TempMonitorFunction)과 해당 구성을 제공하여 Lambda 함수 정의와 초기 버전을 생성합니다. 함수 정의를 생성하려면 먼저 Lambda 함수 가 필요합니다ARN. 함수와 해당 별칭을 생성하려면 Lambda `create-function` 및 `publish-version` 명령을 사용합니다. AWS IoT Greengrass가 Greengrass 그룹 역할에 권한이 지정되어 있기 때문에 해당 역할을 사용하지 않더라도 Lambda의 `create-function` 명령에는 실행 역할ARN의 가 필요합니다. IAM `create-role` 명령을 사용하여 빈 역할을 생성하여 Lambda에서 사용할 ARN를 가져오`create-function`거나 기존 실행 역할을 사용할 수 있습니다.

```
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}}]}"
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS IoT Greengrass 개발자 안내서의 AWS 명령줄 인터페이스를 사용하여 로컬 리소스 액세스를 구성하는 방법을 참조하세요.

- 자세한 API 내용은 명령 참조CreateFunctionDefinition의 섹션을 참조하세요. AWS CLI

### `create-group-certificate-authority`

다음 코드 예시에서는 `create-group-certificate-authority`을 사용하는 방법을 보여 줍니다.

AWS CLI

그룹에 대한 CA(인증 기관)를 생성하려면

다음 `create-group-certificate-authority` 예제에서는 지정된 그룹에 대한 CA를 생성하거나 교대합니다.

```
aws greengrass create-group-certificate-authority \
    --group-id "8eaadd72-ce4b-4f15-892a-0cc4f3a343f1"
```

출력:

```
{
    "GroupCertificateAuthorityArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/groups/8eaadd72-ce4b-4f15-892a-0cc4f3a343f1/certificateauthorities/
d31630d674c4437f6c5dbc0dca56312a902171ce2d086c38e509c8EXAMPLEcc5"
}
```

자세한 내용은 AWS IoT Greengrass 개발자 안내서의 IoT Greengrass 보안을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조CreateGroupCertificateAuthority의 섹션을 참조하세요. AWS CLI

### `create-group-version`

다음 코드 예시에서는 `create-group-version`을 사용하는 방법을 보여 줍니다.

AWS CLI

Greengrass 그룹의 버전을 생성하려면

다음 `create-group-version` 예제에서는 그룹 버전을 생성하고 이를 지정된 그룹과 연결합니다. 버전은 이 그룹 버전에 포함할 엔터티가 포함된 코어, 리소스, 커넥터, 함수 및 구독 버전을 참조합니다. 그룹 버전을 생성하려면 먼저 이러한 엔터티를 생성해야 합니다.

초기 버전으로 리소스 정의를 생성하려면 `create-resource-definition` command.To 초기 버전으로 커넥터 정의를 생성하고, `create-connector-definition` command.To 초기 버전으로 함수 정의를 생성하고, `create-function-definition` command.To 초기 버전으로 구독 정의를 생성하고, `create-subscription-definition` command.To 최신 코어 정의 버전의 ARN를 검색하고, `get-group-version` 명령을 사용하고, 최신 그룹 버전의 ID를 지정합니다.

```
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-
feccbdd69112/versions/aa645c47-ac90-420d-9091-8c7ffa4f103f"
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS IoT Greengrass 개발자 안내서의 IoT Greengrass 그룹 객체 모델 개요를 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조CreateGroupVersion의 섹션을 참조하세요. AWS CLI

### `create-group`

다음 코드 예시에서는 `create-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

Greeengrass 그룹을 생성하려면

다음 `create-group` 예제에서는 라는 그룹을 생성합니다`cli-created-group`.

```
aws greengrass create-group \
    --name cli-created-group
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS IoT Greengrass 개발자 안내서의 IoT Greengrass 그룹 객체 모델 개요를 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조CreateGroup의 섹션을 참조하세요. AWS CLI

### create-logger-definition-version

다음 코드 예시에서는 create-logger-definition-version을 사용하는 방법을 보여 줍니다.

AWS CLI

로거 정의 버전을 생성하려면

다음 create-logger-definition-version 예제에서는 로거 정의 버전을 생성하고 이를 로거 정의와 연결합니다. 버전은 1) 코어 디바이스의 파일 시스템에 대한 시스템 구성 요소 로그, 2) 코어 디바이스의 파일 시스템에 대한 사용자 정의 Lambda 함수 로그, 3) Amazon CloudWatch Logs의 시스템 구성 요소 로그, 4) Amazon CloudWatch Logs의 사용자 정의 Lambda 함수 로그의 네 가지 로깅 구성을 정의합니다. 참고: CloudWatch 로그 통합의 경우 그룹 역할에 적절한 권한을 부여해야 합니다.

```
aws greengrass create-logger-definition-version \
    --logger-definition-id "a454b62a-5d56-4ca9-bdc4-8254e1662cb0" \
    --loggers "[{\"Id\":\"1\",\"Component\":\"GreengrassSystem\",\"Level\":\"ERROR
\",\"Space\":10240,\"Type\":\"FileSystem\"},{\"Id\":\"2\",\"Component\":\"Lambda
\",\"Level\":\"INFO\",\"Space\":10240,\"Type\":\"FileSystem\"},{\"Id\":\"3\",
\"Component\":\"GreengrassSystem\",\"Level\":\"WARN\",\"Type\":\"AWSCloudWatch\"},
{\"Id\":\"4\",\"Component\":\"Lambda\",\"Level\":\"INFO\",\"Type\":\"AWSCloudWatch
\"}]"
```

출력:

```
{
 "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"
}
```

자세한 내용은 AWS IoT Greengrass 개발자 안내서의 IoT Greengrass Logs로 모니터링을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조CreateLoggerDefinitionVersion의 섹션을 참조하세요. AWS CLI

### create-logger-definition

다음 코드 예시에서는 create-logger-definition을 사용하는 방법을 보여 줍니다.

AWS CLI

로거 정의를 생성하려면

다음 `create-logger-definition` 예제에서는 초기 로거 정의 버전이 포함된 로거 정의를 생성합니다. 초기 버전은 1) 코어 디바이스의 파일 시스템에 대한 시스템 구성 요소 로그, 2) 코어 디바이스의 파일 시스템에 대한 사용자 정의 Lambda 함수 로그, 3) Amazon CloudWatch Logs의 사용자 정의 Lambda 함수 로그의 세 가지 로깅 구성을 정의합니다. 참고: CloudWatch 로그 통합의 경우 그룹 역할에 적절한 권한을 부여해야 합니다.

```
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\"}]}"
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS IoT Greengrass 개발자 안내서의 IoT Greengrass Logs로 모니터링을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조CreateLoggerDefinition의 섹션을 참조하세요. AWS CLI

## `create-resource-definition-version`

다음 코드 예시에서는 `create-resource-definition-version`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 정의 버전을 생성하려면

다음 `create-resource-definition-version` 예제에서는 의 새 버전을 생성합니다 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-
ntSlp6\"}}}]"
```

출력:

```
{
    "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"
}
```

- 자세한 API 내용은 명령 참조CreateResourceDefinitionVersion의 섹션을 참조하세요. AWS CLI

### `create-resource-definition`

다음 코드 예시에서는 `create-resource-definition`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 정의를 생성하려면

다음 `create-resource-definition` 예제에서는 Greengrass 그룹에 사용할 리소스 목록을 포함하는 리소스 정의를 생성합니다. 이 예제에서는 리소스 목록을 제공하여 리소스 정의의 초기 버전을 포함합니다. 목록에는 Twilio 권한 부여 토큰에 대한 리소스 하나와 AWS Secrets Manager에 저장된 보안 암호에 ARN 대한 리소스 하나가 포함되어 있습니다. 리소스 정의를 생성하려면 먼저 보안 암호를 생성해야 합니다.

```
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\"}}}]}"
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS IoT Greengrass 개발자 안내서의 AWS 명령줄 인터페이스를 사용하여 로컬 리소스 액세스를 구성하는 방법을 참조하세요.

- 자세한 API 내용은 명령 참조CreateResourceDefinition의 섹션을 참조하세요. AWS CLI

### `create-software-update-job`

다음 코드 예시에서는 `create-software-update-job`을 사용하는 방법을 보여 줍니다.

AWS CLI

코어에 대한 소프트웨어 업데이트 작업을 생성하려면

다음 `create-software-update-job` 예제에서는 이름이 인 over-the-air 코어에서 AWS IoT Greengrass Core 소프트웨어를 업데이트하는 (OTA) 업데이트 작업을 생성합니다`MyFirstGroup_Core`. 이 명령에는 Amazon S3의 소프트웨어 업데이트 패키지에 대한 액세스를 허용하고 를 신뢰할 수 있는 엔터티`iot.amazonaws.com`로 포함하는 IAM 역할이 필요합니다.

```
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
```

출력:

```
{
    "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"
}
```

자세한 내용은 OTA AWS IoT Greengrass 개발자 안내서의 IoT Greengrass Core 소프트웨어 업데이트를 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조CreateSoftwareUpdateJob의 섹션을 참조하세요. AWS CLI

### create-subscription-definition-version

다음 코드 예시에서는 create-subscription-definition-version을 사용하는 방법을 보여 줍니다.

AWS CLI

구독 정의의 새 버전을 생성하려면

다음 create-subscription-definition-version 예제에서는 트리거 알림, 온도 입력 및 출력 상태의 세 가지 구독이 포함된 구독 정의의 새 버전을 생성합니다.

```
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\"}]"
```

출력:

```
{
    "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"
}
```

- 자세한 API 내용은 명령 참조CreateSubscriptionDefinitionVersion의 섹션을 참조하세요. AWS CLI

### create-subscription-definition

다음 코드 예시에서는 create-subscription-definition을 사용하는 방법을 보여 줍니다.

AWS CLI

구독 정의를 생성하려면

다음 create-subscription-definition 예제에서는 구독 정의를 생성하고 초기 버전을 지정합니다. 초기 버전에는 세 가지 구독이 포함되어 있습니다. 하나는 커넥터가 구독하는 MQTT 주제에 대한 구독이고, 다른 하나는 함수가 AWS IoT 에서 온도 판독값을 수신할 수 있도록 허용하고, 다른 하나는 AWS IoT가 커넥터에서 상태 정보를 수신할 수 있도록 허용합니다. 이 예제에서는 Lambda의 create-alias 명령을 사용하여 이전에 생성된 Lambda 함수 별칭에 ARN 대한 를 제공합니다.

```
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\"}]}"
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS IoT Greengrass 개발자 안내서의 커넥터 시작하기(CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조CreateSubscriptionDefinition의 섹션을 참조하세요. AWS CLI

### delete-connector-definition

다음 코드 예시에서는 delete-connector-definition을 사용하는 방법을 보여 줍니다.

AWS CLI

커넥터 정의를 삭제하려면

다음 delete-connector-definition 예제에서는 지정된 Greengrass 커넥터 정의를 삭제합니다. 그룹에서 사용하는 커넥터 정의를 삭제하면 해당 그룹을 성공적으로 배포할 수 없습니다.

```
aws greengrass delete-connector-definition \
    --connector-definition-id "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8"
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeleteConnectorDefinition의 섹션을 참조하세요. AWS CLI

### delete-core-definition

다음 코드 예시에서는 delete-core-definition을 사용하는 방법을 보여 줍니다.

AWS CLI

코어 정의를 삭제하려면

다음 delete-core-definition 예제에서는 모든 버전을 포함하여 지정된 Greengrass 코어 정의를 삭제합니다. Greengrass 그룹과 연결된 코어를 삭제하면 해당 그룹을 성공적으로 배포할 수 없습니다.

```
aws greengrass delete-core-definition \
    --core-definition-id "ff36cc5f-9f98-4994-b468-9d9b6dc52abd"
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeleteCoreDefinition의 섹션을 참조하세요. AWS CLI

### delete-device-definition

다음 코드 예시에서는 delete-device-definition을 사용하는 방법을 보여 줍니다.

AWS CLI

디바이스 정의를 삭제하려면

다음 delete-device-definition 예제에서는 모든 버전을 포함하여 지정된 디바이스 정의를 삭제합니다. 그룹 버전에서 사용하는 디바이스 정의 버전을 삭제하면 그룹 버전을 성공적으로 배포할 수 없습니다.

```
aws greengrass delete-device-definition \
    --device-definition-id "f9ba083d-5ad4-4534-9f86-026a45df1ccd"
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeleteDeviceDefinition의 섹션을 참조하세요. AWS CLI

### delete-function-definition

다음 코드 예시에서는 delete-function-definition을 사용하는 방법을 보여 줍니다.

AWS CLI

함수 정의를 삭제하려면

다음 delete-function-definition 예제에서는 지정된 Greengrass 함수 정의를 삭제합니다. 그룹에서 사용하는 함수 정의를 삭제하면 해당 그룹을 성공적으로 배포할 수 없습니다.

```
aws greengrass delete-function-definition \
    --function-definition-id "fd4b906a-dff3-4c1b-96eb-52ebfcfac06a"
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeleteFunctionDefinition의 섹션을 참조하세요. AWS CLI

### delete-group

다음 코드 예시에서는 delete-group을 사용하는 방법을 보여 줍니다.

AWS CLI

그룹을 삭제하려면

다음 delete-group 예제에서는 지정된 Greengrass 그룹을 삭제합니다.

```
aws greengrass delete-group \
    --group-id "4e22bd92-898c-436b-ade5-434d883ff749"
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeleteGroup의 섹션을 참조하세요. AWS CLI

### delete-logger-definition

다음 코드 예시에서는 delete-logger-definition을 사용하는 방법을 보여 줍니다.

AWS CLI

로거 정의를 삭제하려면

다음 delete-logger-definition 예제에서는 모든 로거 정의 버전을 포함하여 지정된 로거 정의를 삭제합니다. 그룹 버전에서 사용되는 로거 정의 버전을 삭제하면 그룹 버전을 성공적으로 배포할 수 없습니다.

```
aws greengrass delete-logger-definition \
```

```
    --logger-definition-id "a454b62a-5d56-4ca9-bdc4-8254e1662cb0"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT Greengrass 개발자 안내서의 IoT Greengrass Logs로 모니터링을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조DeleteLoggerDefinition의 섹션을 참조하세요. AWS CLI

### delete-resource-definition

다음 코드 예시에서는 delete-resource-definition을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 정의를 삭제하려면

다음 delete-resource-definition 예제에서는 모든 리소스 버전을 포함하여 지정된 리소스 정의를 삭제합니다. 그룹에서 사용하는 리소스 정의를 삭제하면 해당 그룹을 성공적으로 배포할 수 없습니다.

```
aws greengrass delete-resource-definition \
    --resource-definition-id "ad8c101d-8109-4b0e-b97d-9cc5802ab658"
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeleteResourceDefinition의 섹션을 참조하세요. AWS CLI

### delete-subscription-definition

다음 코드 예시에서는 delete-subscription-definition을 사용하는 방법을 보여 줍니다.

AWS CLI

구독 정의를 삭제하려면

다음 delete-subscription-definition 예제에서는 지정된 Greengrass 구독 정의를 삭제합니다. 그룹에서 사용 중인 구독을 삭제하면 해당 그룹을 성공적으로 배포할 수 없습니다.

```
aws greengrass delete-subscription-definition \
    --subscription-definition-id "cd6f1c37-d9a4-4e90-be94-01a7404f5967"
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeleteSubscriptionDefinition의 섹션을 참조하세요. AWS CLI

### `disassociate-role-from-group`

다음 코드 예시에서는 `disassociate-role-from-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

Greengrass 그룹에서 역할을 연결 해제하려면

다음 `disassociate-role-from-group` 예제에서는 지정된 Greengrass 그룹에서 IAM 역할을 연결 해제합니다.

```
aws greengrass disassociate-role-from-group \
    --group-id 2494ee3f-7f8a-4e92-a78b-d205f808b84b
```

출력:

```
{
    "DisassociatedAt": "2019-09-10T20:05:49Z"
}
```

자세한 내용은 AWS IoT Greengrass 개발자 안내서의 그룹 역할 구성을 참조하세요.

- 자세한 API 내용은 명령 참조DisassociateRoleFromGroup의 섹션을 참조하세요. AWS CLI

### `disassociate-service-role-from-account`

다음 코드 예시에서는 `disassociate-service-role-from-account`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 계정에서 서비스 역할을 연결 해제하려면

다음 `disassociate-service-role-from-account` 예제에서는 AWS 계정과 연결된 서비스 역할을 제거합니다. AWS 리전에서 서비스 역할을 사용하지 않는 경우 `delete-role-policy` 명령을 사용하여 `AWSGreengrassResourceAccessRolePolicy` 관리형 정책을 역할에서 분리한 다음 `delete-role` 명령을 사용하여 역할을 삭제합니다.

```
aws greengrass disassociate-service-role-from-account
```

출력:

```
{
    "DisassociatedAt": "2019-06-25T22:12:55Z"
}
```

자세한 내용은 AWS IoT Greengrass 개발자 안내서의 Greengrass 서비스 역할을 참조하세요.

- 자세한 API 내용은 명령 참조DisassociateServiceRoleFromAccount의 섹션을 참조하세요. AWS CLI

### get-associated-role

다음 코드 예시에서는 get-associated-role을 사용하는 방법을 보여 줍니다.

AWS CLI

Greengrass 그룹과 연결된 역할을 가져오려면

다음 get-associated-role 예제에서는 지정된 Greengrass 그룹과 연결된 IAM 역할을 가져옵니다. 그룹 역할은 로컬 Lambda 함수 및 커넥터에서 AWS 서비스에 액세스하는 데 사용됩니다.

```
aws greengrass get-associated-role \
    --group-id 2494ee3f-7f8a-4e92-a78b-d205f808b84b
```

출력:

```
{
    "RoleArn": "arn:aws:iam::123456789012:role/GG-Group-Role",
    "AssociatedAt": "2019-09-10T20:03:30Z"
}
```

자세한 내용은 AWS IoT Greengrass 개발자 안내서의 그룹 역할 구성을 참조하세요.

- 자세한 API 내용은 명령 참조GetAssociatedRole의 섹션을 참조하세요. AWS CLI

### get-bulk-deployment-status

다음 코드 예시에서는 get-bulk-deployment-status을 사용하는 방법을 보여 줍니다.

AWS CLI

대량 배포 상태를 확인하려면

다음 `get-bulk-deployment-status` 예제에서는 지정된 대량 배포 작업에 대한 상태 정보를 검색합니다. 이 예제에서는 배포할 그룹을 지정한 파일에 잘못된 입력 레코드가 있습니다.

```
aws greengrass get-bulk-deployment-status \
    --bulk-deployment-id "870fb41b-6288-4e0c-bc76-a7ba4b4d3267"
```

출력:

```
{
    "BulkDeploymentMetrics": {
        "InvalidInputRecords": 1,
        "RecordsProcessed": 1,
        "RetryAttempts": 0
    },
    "BulkDeploymentStatus": "Completed",
    "CreatedAt": "2019-06-25T16:11:33.265Z",
    "tags": {}
}
```

자세한 내용은 AWS IoT Greengrass 개발자 안내서의 그룹에 대한 대량 배포 생성을 참조하세요.

- 자세한 API 내용은 명령 참조GetBulkDeploymentStatus의 섹션을 참조하세요. AWS CLI

## get-connectivity-info

다음 코드 예시에서는 `get-connectivity-info`을 사용하는 방법을 보여 줍니다.

AWS CLI

Greengrass 코어의 연결 정보를 가져오려면

다음 `get-connectivity-info` 예제에서는 디바이스가 지정된 Greengrass 코어에 연결하는 데 사용할 수 있는 엔드포인트를 보여줍니다. 연결 정보는 IP 주소 또는 도메인 이름의 목록으로, 해당 포트 번호 및 선택적 고객 정의 메타데이터를 포함합니다.

```
aws greengrass get-connectivity-info \
```

```
    --thing-name "MyGroup_Core"
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetConnectivityInfo의 섹션을 참조하세요. AWS CLI

### `get-connector-definition-version`

다음 코드 예시에서는 `get-connector-definition-version`을 사용하는 방법을 보여 줍니다.

AWS CLI

커넥터 정의의 특정 버전에 대한 정보를 검색하려면

다음 get-connector-definition-version 예제에서는 지정된 커넥터 정의의 지정된 버전에 대한 정보를 검색합니다. 커넥터 정의IDs의 모든 버전의 를 검색하려면 list-connector-definition-versions 명령을 사용합니다. 커넥터 정의에 추가된 마지막 버전의 ID를 검색하려면 get-connector-definition 명령을 사용하여 LatestVersion 속성을 확인합니다.

```
aws greengrass get-connector-definition-version \
    --connector-definition-id "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8" \
    --connector-definition-version-id "63c57963-c7c2-4a26-a7e2-7bf478ea2623"
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS IoT Greengrass 개발자 안내서의 Greengrass 커넥터를 사용하여 서비스 및 프로토콜과 통합을 참조하세요.

- 자세한 API 내용은 명령 참조GetConnectorDefinitionVersion의 섹션을 참조하세요. AWS CLI

### get-connector-definition

다음 코드 예시에서는 get-connector-definition을 사용하는 방법을 보여 줍니다.

AWS CLI

커넥터 정의에 대한 정보를 검색하려면

다음 `get-connector-definition` 예제에서는 지정된 커넥터 정의에 대한 정보를 검색합니다. 커넥터 정의IDs의 를 검색하려면 `list-connector-definitions` 명령을 사용합니다.

```
aws greengrass get-connector-definition \
    --connector-definition-id "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8"
```

출력:

```
{
    "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": {}
}
```

자세한 내용은 AWS IoT Greengrass 개발자 안내서의 Greengrass 커넥터를 사용하여 서비스 및 프로토콜과 통합을 참조하세요.

- 자세한 API 내용은 명령 참조GetConnectorDefinition의 섹션을 참조하세요. AWS CLI

### `get-core-definition-version`

다음 코드 예시에서는 `get-core-definition-version`을 사용하는 방법을 보여 줍니다.

AWS CLI

Greengrass 코어 정의의 특정 버전에 대한 세부 정보를 검색하려면

다음 `get-core-definition-version` 예제에서는 지정된 코어 정의의 지정된 버전에 대한 정보를 검색합니다. 코어 정의IDs의 모든 버전의 를 검색하려면 `list-core-definition-`

versions 명령을 사용합니다. 코어 정의에 추가된 마지막 버전의 ID를 검색하려면 get-core-definition 명령을 사용하여 LatestVersion 속성을 확인합니다.

```
aws greengrass get-core-definition-version \
    --core-definition-id "c906ed39-a1e3-4822-a981-7b9bd57b4b46"  \
    --core-definition-version-id "42aeeac3-fd9d-4312-a8fd-ffa9404a20e0"
```

출력:

```
{
    "Arn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/definition/cores/
c906ed39-a1e3-4822-a981-7b9bd57b4b46/versions/42aeeac3-fd9d-4312-a8fd-ffa9404a20e0",
    "CreationTimestamp": "2019-06-18T16:21:21.351Z",
    "Definition": {
        "Cores": [
            {
                "CertificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/928dea7b82331b47c3ff77b0e763fc5e64e2f7c884e6ef391baed9b6b8e21b45",
                "Id": "1a39aac7-0885-4417-91f6-23e4cea6c511",
                "SyncShadow": false,
                "ThingArn": "arn:aws:iot:us-west-2:123456789012:thing/
GGGroup4Pi3_Core"
            }
        ]
    },
    "Id": "c906ed39-a1e3-4822-a981-7b9bd57b4b46",
    "Version": "42aeeac3-fd9d-4312-a8fd-ffa9404a20e0"
}
```

- 자세한 API 내용은 명령 참조GetCoreDefinitionVersion의 섹션을 참조하세요. AWS CLI

### get-core-definition

다음 코드 예시에서는 get-core-definition을 사용하는 방법을 보여 줍니다.

AWS CLI

Greengrass 코어 정의에 대한 세부 정보를 검색하려면

다음 get-core-definition 예제에서는 지정된 코어 정의에 대한 정보를 검색합니다. 코어 정의 IDs의 를 검색하려면 list-core-definitions 명령을 사용합니다.

```
aws greengrass get-core-definition \
    --core-definition-id "c906ed39-a1e3-4822-a981-7b9bd57b4b46"
```

출력:

```
{
    "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": {}
}
```

- 자세한 API 내용은 명령 참조GetCoreDefinition의 섹션을 참조하세요. AWS CLI

### get-deployment-status

다음 코드 예시에서는 get-deployment-status을 사용하는 방법을 보여 줍니다.

AWS CLI

배포 상태를 검색하려면

다음 get-deployment-status 예제에서는 지정된 Greengrass 그룹의 지정된 배포 상태를 검색합니다. 배포 ID를 가져오려면 list-deployments 명령을 사용하고 그룹 ID를 지정합니다.

```
aws greengrass get-deployment-status \
    --group-id "1013db12-8b58-45ff-acc7-704248f66731" \
    --deployment-id "1065b8a0-812b-4f21-9d5d-e89b232a530f"
```

출력:

```
{
    "DeploymentStatus": "Success",
    "DeploymentType": "NewDeployment",
    "UpdatedAt": "2019-06-18T17:04:44.761Z"
```

```
}
```

- 자세한 API 내용은 명령 참조GetDeploymentStatus의 섹션을 참조하세요. AWS CLI

### get-device-definition-version

다음 코드 예시에서는 get-device-definition-version을 사용하는 방법을 보여 줍니다.

AWS CLI

디바이스 정의 버전을 가져오려면

다음 get-device-definition-version 예제에서는 지정된 디바이스 정의의 지정된 버전에 대한 정보를 검색합니다. 디바이스 정의IDs의 모든 버전의 를 검색하려면 list-device-definition-versions 명령을 사용합니다. 디바이스 정의에 추가된 마지막 버전의 ID를 검색하려면 get-device-definition 명령을 사용하여 LatestVersion 속성을 확인합니다.

```
aws greengrass get-device-definition-version \
    --device-definition-id "f9ba083d-5ad4-4534-9f86-026a45df1ccd" \
    --device-definition-version-id "83c13984-6fed-447e-84d5-5b8aa45d5f71"
```

출력:

```
{
    "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"
}
```

- 자세한 API 내용은 명령 참조GetDeviceDefinitionVersion의 섹션을 참조하세요. AWS CLI

### get-device-definition

다음 코드 예시에서는 get-device-definition을 사용하는 방법을 보여 줍니다.

AWS CLI

디바이스 정의를 가져오려면

다음 get-device-definition 예제에서는 지정된 디바이스 정의에 대한 정보를 검색합니다. 디바이스 정의IDs의 를 검색하려면 list-device-definitions 명령을 사용합니다.

```
aws greengrass get-device-definition \
    --device-definition-id "f9ba083d-5ad4-4534-9f86-026a45df1ccd"
```

출력:

```
{
    "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"
}
```

- 자세한 API 내용은 명령 참조GetDeviceDefinition의 섹션을 참조하세요. AWS CLI

### get-function-definition-version

다음 코드 예시에서는 get-function-definition-version을 사용하는 방법을 보여 줍니다.

AWS CLI

Lambda 함수의 특정 버전에 대한 세부 정보를 검색하려면

다음은 지정된 함수 정의의 지정된 버전에 대한 정보를 get-function-definition-version 검색합니다. 함수 정의IDs의 모든 버전의 를 검색하려면 list-function-definition-versions 명령을 사용합니다. 함수 정의에 추가된 마지막 버전의 ID를 검색하려면 get-function-definition 명령을 사용하고 LatestVersion 속성을 확인합니다.

```
aws greengrass get-function-definition-version \
    --function-definition-id "063f5d1a-1dd1-40b4-9b51-56f8993d0f85" \
    --function-definition-version-id "9748fda7-1589-4fcc-ac94-f5559e88678b"
```

출력:

```
{
    "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"
}
```

- 자세한 API 내용은 명령 참조GetFunctionDefinitionVersion의 섹션을 참조하세요. AWS CLI

### get-function-definition

다음 코드 예시에서는 get-function-definition을 사용하는 방법을 보여 줍니다.

AWS CLI

함수 정의를 검색하려면

다음 get-function-definition 예제에서는 지정된 함수 정의에 대한 세부 정보를 표시합니다. 함수 정의IDs의 를 검색하려면 list-function-definitions 명령을 사용합니다.

```
aws greengrass get-function-definition \
    --function-definition-id "063f5d1a-1dd1-40b4-9b51-56f8993d0f85"
```

출력:

```
{
    "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": {}
}
```

- 자세한 API 내용은 명령 참조GetFunctionDefinition의 섹션을 참조하세요. AWS CLI

### get-group-certificate-authority

다음 코드 예시에서는 get-group-certificate-authority을 사용하는 방법을 보여 줍니다.

AWS CLI

Greengrass 그룹과 연결된 CA를 검색하려면

다음 get-group-certificate-authority 예제에서는 지정된 Greengrass 그룹과 연결된 인증 기관(CA)을 검색합니다. 인증서 기관 ID를 가져오려면 list-group-certificate-authorities 명령을 사용하고 그룹 ID를 지정합니다.

```
aws greengrass get-group-certificate-authority \
    --group-id "1013db12-8b58-45ff-acc7-704248f66731" \
    --certificate-authority-
id "f0430e1736ea8ed30cc5d5de9af67a7e3586bad9ae4d89c2a44163f65fdd8cf6"
```

출력:

```
{
    "GroupCertificateAuthorityArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/groups/1013db12-8b58-45ff-acc7-704248f66731/certificateauthorities/
f0430e1736ea8ed30cc5d5de9af67a7e3586bad9ae4d89c2a44163f65fdd8cf6",
    "GroupCertificateAuthorityId":
 "f0430e1736ea8ed30cc5d5de9af67a7e3586bad9ae4d89c2a44163f65fdd8cf6",
    "PemEncodedCertificate": "-----BEGIN CERTIFICATE-----
MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBWEXAMPLEGA1UEBhMC
VVMxCzAJBgNVBAgTAldBMRAwDEXAMPLEEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAEXAMPLESBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jEXAMPLENMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0EXAMPLEBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWEXAMPLEDASBgNVBAsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWEXAMPLEgkqhkiG9w0BCQEWEG5vb25lQGFt
YXpvbi5EXAMPLE8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
```

```
21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CEXAMPLE93vUEIO3IyNoH/f0wYK8m9T
rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswYEXAMPLEgpE
Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKEXAMPLEAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----\n"
}
```

- 자세한 API 내용은 명령 참조GetGroupCertificateAuthority의 섹션을 참조하세요. AWS CLI

### get-group-certificate-configuration

다음 코드 예시에서는 get-group-certificate-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

Greengrass 그룹에서 사용하는 인증 기관의 구성을 검색하려면

다음 get-group-certificate-configuration 예제에서는 지정된 Greengrass 그룹에서 사용하는 인증 기관(CA)에 대한 구성을 검색합니다.

```
aws greengrass get-group-certificate-configuration \
    --group-id "1013db12-8b58-45ff-acc7-704248f66731"
```

출력:

```
{
    "CertificateAuthorityExpiryInMilliseconds": 2524607999000,
    "CertificateExpiryInMilliseconds": 604800000,
    "GroupId": "1013db12-8b58-45ff-acc7-704248f66731"
}
```

- 자세한 API 내용은 명령 참조GetGroupCertificateConfiguration의 섹션을 참조하세요. AWS CLI

### get-group-version

다음 코드 예시에서는 get-group-version을 사용하는 방법을 보여 줍니다.

AWS CLI

Greengrass 그룹 버전에 대한 정보를 검색하려면

다음 get-group-version 예제에서는 지정된 그룹의 지정된 버전에 대한 정보를 검색합니다. 그룹의 모든 버전의 IDs 를 검색하려면 list-group-versions 명령을 사용합니다. 그룹에 추가된 마지막 버전의 ID를 검색하려면 get-group 명령을 사용하여 LatestVersion 속성을 확인합니다.

```
aws greengrass get-group-version \
    --group-id "1013db12-8b58-45ff-acc7-704248f66731"  \
    --group-version-id "115136b3-cfd7-4462-b77f-8741a4b00e5e"
```

출력:

```
{
    "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"
}
```

- 자세한 API 내용은 명령 참조GetGroupVersion의 섹션을 참조하세요. AWS CLI

**get-group**

다음 코드 예시에서는 get-group을 사용하는 방법을 보여 줍니다.

AWS CLI

Greengrass 그룹에 대한 정보를 검색하려면

다음 `get-group` 예제에서는 지정된 Greengrass 그룹에 대한 정보를 검색합니다. 그룹의 IDs 를 검색하려면 `list-groups` 명령을 사용합니다.

```
aws greengrass get-group \
    --group-id "1013db12-8b58-45ff-acc7-704248f66731"
```

출력:

```
{
    "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": {}
}
```

- 자세한 API 내용은 명령 참조GetGroup의 섹션을 참조하세요. AWS CLI

### get-logger-definition-version

다음 코드 예시에서는 `get-logger-definition-version`을 사용하는 방법을 보여 줍니다.

AWS CLI

로거 정의 버전에 대한 정보를 검색하려면

다음 `get-logger-definition-version` 예제에서는 지정된 로거 정의의 지정된 버전에 대한 정보를 검색합니다. 로거 정의IDs의 모든 버전의 를 검색하려면 `list-logger-definition-versions` 명령을 사용합니다. 로거 정의에 추가된 마지막 버전의 ID를 검색하려면 `get-logger-definition` 명령을 사용하고 `LatestVersion` 속성을 확인합니다.

```
aws greengrass get-logger-definition-version \
```

```
    --logger-definition-id "49eeeb66-f1d3-4e34-86e3-3617262abf23" \
    --logger-definition-version-id "5e3f6f64-a565-491e-8de0-3c0d8e0f2073"
```

출력:

```
{
    "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"
}
```

- 자세한 API 내용은 명령 참조GetLoggerDefinitionVersion의 섹션을 참조하세요. AWS CLI

### get-logger-definition

다음 코드 예시에서는 get-logger-definition을 사용하는 방법을 보여 줍니다.

AWS CLI

로거 정의에 대한 정보를 검색하려면

다음 get-logger-definition 예제에서는 지정된 로거 정의에 대한 정보를 검색합니다. 로거 정의IDs의 를 검색하려면 list-logger-definitions 명령을 사용합니다.

```
aws greengrass get-logger-definition \
    --logger-definition-id "49eeeb66-f1d3-4e34-86e3-3617262abf23"
```

출력:

```
{
    "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": {}
}
```

- 자세한 API 내용은 명령 참조GetLoggerDefinition의 섹션을 참조하세요. AWS CLI

### get-resource-definition-version

다음 코드 예시에서는 get-resource-definition-version을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 정의의 특정 버전에 대한 정보를 검색하려면

다음 get-resource-definition-version 예제에서는 지정된 리소스 정의의 지정된 버전에 대한 정보를 검색합니다. 리소스 정의IDs의 모든 버전의 를 검색하려면 list-resource-definition-versions 명령을 사용합니다. 리소스 정의에 추가된 마지막 버전의 ID를 검색하려면 get-resource-definition 명령을 사용하고 LatestVersion 속성을 확인합니다.

```
aws greengrass get-resource-definition-version \
    --resource-definition-id "ad8c101d-8109-4b0e-b97d-9cc5802ab658" \
    --resource-definition-version-id "26e8829a-491a-464d-9c87-664bf6f6f2be"
```

출력:

```
{
    "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"
}
```

- 자세한 API 내용은 명령 참조GetResourceDefinitionVersion의 섹션을 참조하세요. AWS CLI

### `get-resource-definition`

다음 코드 예시에서는 `get-resource-definition`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 정의에 대한 정보를 검색하려면

다음 `get-resource-definition` 예제에서는 지정된 리소스 정의에 대한 정보를 검색합니다. 리소스 정의IDs의 를 검색하려면 `list-resource-definitions` 명령을 사용합니다.

```
aws greengrass get-resource-definition \
    --resource-definition-id "ad8c101d-8109-4b0e-b97d-9cc5802ab658"
```

출력:

```
{
    "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": {}
}
```

- 자세한 API 내용은 명령 참조GetResourceDefinition의 섹션을 참조하세요. AWS CLI

### get-service-role-for-account

다음 코드 예시에서는 get-service-role-for-account을 사용하는 방법을 보여 줍니다.

AWS CLI

계정에 연결된 서비스 역할의 세부 정보를 검색하려면

다음 get-service-role-for-account 예제에서는 AWS 계정에 연결된 서비스 역할에 대한 정보를 검색합니다.

```
aws greengrass get-service-role-for-account
```

출력:

```
{
    "AssociatedAt": "2018-10-18T15:59:20Z",
    "RoleArn": "arn:aws:iam::123456789012:role/service-role/Greengrass_ServiceRole"
}
```

자세한 내용은 AWS IoT Greengrass 개발자 안내서의 Greengrass 서비스 역할을 참조하세요.

- 자세한 API 내용은 명령 참조GetServiceRoleForAccount의 섹션을 참조하세요. AWS CLI

### get-subscription-definition-version

다음 코드 예시에서는 get-subscription-definition-version을 사용하는 방법을 보여 줍니다.

AWS CLI

구독 정의의 특정 버전에 대한 정보를 검색하려면

다음 get-subscription-definition-version 예제에서는 지정된 구독 정의의 지정된 버전에 대한 정보를 검색합니다. 구독 정의IDs의 모든 버전의 를 검색하려면 list-subscription-definition-versions 명령을 사용합니다. 구독 정의에 추가된 마지막 버전의 ID를 검색하려면 get-subscription-definition 명령을 사용하여 LatestVersion 속성을 확인합니다.

```
aws greengrass get-subscription-definition-version \
```

```
    --subscription-definition-id "70e49321-83d5-45d2-bc09-81f4917ae152" \
    --subscription-definition-version-id "88ae8699-12ac-4663-ba3f-4d7f0519140b"
```

출력:

```
{
    "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"
}
```

- 자세한 API 내용은 명령 참조GetSubscriptionDefinitionVersion의 섹션을 참조하세요. AWS CLI

### get-subscription-definition

다음 코드 예시에서는 get-subscription-definition을 사용하는 방법을 보여 줍니다.

AWS CLI

구독 정의에 대한 정보를 검색하려면

다음 get-subscription-definition 예제에서는 지정된 구독 정의에 대한 정보를 검색합니다. 구독 정의IDs의 를 검색하려면 list-subscription-definitions 명령을 사용합니다.

```
aws greengrass get-subscription-definition \
    --subscription-definition-id "70e49321-83d5-45d2-bc09-81f4917ae152"
```

출력:

```
{
    "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": {}
}
```

- 자세한 API 내용은 명령 참조GetSubscriptionDefinition의 섹션을 참조하세요. AWS CLI

### `get-thing-runtime-configuration`

다음 코드 예시에서는 `get-thing-runtime-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

Greengrass 코어의 런타임 구성을 검색하려면

다음 `get-thing-runtime-configuration` 예제에서는 Greengrass 코어의 런타임 구성을 검색합니다. 런타임 구성을 검색하려면 먼저 `update-thing-runtime-configuration` 명령을 사용하여 코어에 대한 런타임 구성을 생성해야 합니다.

```
aws greengrass get-thing-runtime-configuration \
    --thing-name SampleGreengrassCore
```

출력:

```
{
    "RuntimeConfiguration": {
        "TelemetryConfiguration": {
            "ConfigurationSyncStatus": "OutOfSync",
            "Telemetry": "On"
        }
    }
}
```

자세한 내용은 AWS IoT Greengrass 개발자 안내서의 원격 측정 설정 구성을 참조하세요.

- 자세한 API 내용은 명령 참조GetThingRuntimeConfiguration의 섹션을 참조하세요. AWS CLI

### `list-bulk-deployment-detailed-reports`

다음 코드 예시에서는 `list-bulk-deployment-detailed-reports`을 사용하는 방법을 보여 줍니다.

AWS CLI

대량 배포의 개별 배포에 대한 정보를 나열하려면

다음 `list-bulk-deployment-detailed-reports` 예제에서는 상태를 포함하여 대량 배포 작업의 개별 배포에 대한 정보를 표시합니다.

```
aws greengrass list-bulk-deployment-detailed-reports \
    --bulk-deployment-id 42ce9c42-489b-4ed4-b905-8996aa50ef9d
```

출력:

```
{
    "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"
        },
        ...
    ]
}
```

자세한 내용은 AWS IoT Greengrass 개발자 안내서의 그룹에 대한 대량 배포 생성을 참조하세요.

- 자세한 API 내용은 명령 참조ListBulkDeploymentDetailedReports의 섹션을 참조하세요. AWS CLI

### `list-bulk-deployments`

다음 코드 예시에서는 `list-bulk-deployments`을 사용하는 방법을 보여 줍니다.

AWS CLI

대량 배포를 나열하려면

다음 `list-bulk-deployments` 예제에서는 모든 대량 배포를 나열합니다.

```
aws greengrass list-bulk-deployments
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS IoT Greengrass 개발자 안내서의 그룹에 대한 대량 배포 생성을 참조하세요.

- 자세한 API 내용은 명령 참조ListBulkDeployments의 섹션을 참조하세요. AWS CLI

### list-connector-definition-versions

다음 코드 예시에서는 list-connector-definition-versions을 사용하는 방법을 보여 줍니다.

AWS CLI

커넥터 정의에 사용할 수 있는 버전을 나열하려면

다음 list-connector-definition-versions 예제에서는 지정된 커넥터 정의에 사용할 수 있는 버전을 나열합니다. list-connector-definitions 명령을 사용하여 커넥터 정의 ID를 가져옵니다.

```
aws greengrass list-connector-definition-versions \
    --connector-definition-id "b5c4ebfd-f672-49a3-83cd-31c7216a7bb8"
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS IoT Greengrass 개발자 안내서의 Greengrass 커넥터를 사용하여 서비스 및 프로토콜과 통합을 참조하세요.

- 자세한 API 내용은 명령 참조ListConnectorDefinitionVersions의 섹션을 참조하세요. AWS CLI

### list-connector-definitions

다음 코드 예시에서는 list-connector-definitions을 사용하는 방법을 보여 줍니다.

AWS CLI

정의된 Greengrass 커넥터를 나열하려면

다음 `list-connector-definitions` 예제에서는 AWS 계정에 정의된 모든 Greengrass 커넥터를 나열합니다.

```
aws greengrass list-connector-definitions
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS IoT Greengrass 개발자 안내서의 Greengrass 커넥터를 사용하여 서비스 및 프로토콜과 통합을 참조하세요.

- 자세한 API 내용은 명령 참조ListConnectorDefinitions의 섹션을 참조하세요. AWS CLI

### `list-core-definition-versions`

다음 코드 예시에서는 `list-core-definition-versions`을 사용하는 방법을 보여 줍니다.

AWS CLI

Greengrass 코어 정의의 버전을 나열하려면

다음 `list-core-definitions` 예제에서는 지정된 Greengrass 코어 정의의 모든 버전을 나열합니다. `list-core-definitions` 명령을 사용하여 버전 ID를 가져올 수 있습니다.

```
aws greengrass list-core-definition-versions \
```

```
    --core-definition-id "eaf280cb-138c-4d15-af36-6f681a1348f7"
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListCoreDefinitionVersions의 섹션을 참조하세요. AWS CLI

### `list-core-definitions`

다음 코드 예시에서는 `list-core-definitions`을 사용하는 방법을 보여 줍니다.

AWS CLI

Greengrass 코어 정의를 나열하려면

다음 `list-core-definitions` 예제에서는 AWS 계정의 모든 Greengrass 코어 정의를 나열합니다.

```
aws greengrass list-core-definitions
```

출력:

```
{
    "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": "bcdf9e86-3793-491e-93af-3cdfbf4e22b7",
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/cores/0507843c-c1ef-4f06-b051-817030df7e7d/versions/
bcdf9e86-3793-491e-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": "42aeeac3-fd9d-4312-a8fd-ffa9404a20e0",
            "LatestVersionArn": "arn:aws:greengrass:us-west-2:123456789012:/
greengrass/definition/cores/c906ed39-a1e3-4822-a981-7b9bd57b4b46/versions/42aeeac3-
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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListCoreDefinitions의 섹션을 참조하세요. AWS CLI

### list-deployments

다음 코드 예시에서는 list-deployments을 사용하는 방법을 보여 줍니다.

AWS CLI

Greengrass 그룹의 배포를 나열하려면

다음 list-deployments 예제에서는 지정된 Greengrass 그룹에 대한 배포를 나열합니다. list-groups 명령을 사용하여 그룹 ID를 조회할 수 있습니다.

```
aws greengrass list-deployments \
    --group-id "1013db12-8b58-45ff-acc7-704248f66731"
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListDeployments의 섹션을 참조하세요. AWS CLI

### list-device-definition-versions

다음 코드 예시에서는 list-device-definition-versions을 사용하는 방법을 보여 줍니다.

AWS CLI

디바이스 정의의 버전을 나열하려면

다음 list-device-definition-versions 예제에서는 지정된 디바이스 정의와 연결된 디바이스 정의 버전을 보여줍니다.

```
aws greengrass list-device-definition-versions \
    --device-definition-id "f9ba083d-5ad4-4534-9f86-026a45df1ccd"
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListDeviceDefinitionVersions의 섹션을 참조하세요. AWS CLI

### `list-device-definitions`

다음 코드 예시에서는 `list-device-definitions`을 사용하는 방법을 보여 줍니다.

AWS CLI

디바이스 정의를 나열하려면

다음 `list-device-definitions` 예제에서는 지정된 AWS 리전의 AWS 계정에서 디바이스 정의에 대한 세부 정보를 표시합니다.

```
aws greengrass list-device-definitions \
    --region us-west-2
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListDeviceDefinitions의 섹션을 참조하세요. AWS CLI

### list-function-definition-versions

다음 코드 예시에서는 list-function-definition-versions을 사용하는 방법을 보여 줍니다.

AWS CLI

Lambda 함수의 버전을 나열하려면

다음 list-function-definition-versions 예제에서는 지정된 Lambda 함수의 모든 버전을 나열합니다. list-function-definitions 명령을 사용하여 ID를 가져올 수 있습니다.

```
aws greengrass list-function-definition-versions \
    --function-definition-id "063f5d1a-1dd1-40b4-9b51-56f8993d0f85"
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListFunctionDefinitionVersions의 섹션을 참조하세요. AWS CLI

### `list-function-definitions`

다음 코드 예시에서는 `list-function-definitions`을 사용하는 방법을 보여 줍니다.

AWS CLI

Lambda 함수를 나열하려면

다음 `list-function-definitions` 예제에서는 AWS 계정에 정의된 모든 Lambda 함수를 나열합니다.

```
aws greengrass list-function-definitions
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListFunctionDefinitions의 섹션을 참조하세요. AWS CLI

### `list-group-certificate-authorities`

다음 코드 예시에서는 `list-group-certificate-authorities`을 사용하는 방법을 보여 줍니다.

AWS CLI

그룹의 현재CAs를 나열하려면

다음 `list-group-certificate-authorities` 예제에서는 지정된 Greengrass 그룹의 현재 인증 기관(CAs)을 나열합니다.

```
aws greengrass list-group-certificate-authorities \
    --group-id "1013db12-8b58-45ff-acc7-704248f66731"
```

출력:

```
{
    "GroupCertificateAuthorities": [
        {
            "GroupCertificateAuthorityArn": "arn:aws:greengrass:us-
west-2:123456789012:/greengrass/groups/1013db12-8b58-45ff-acc7-704248f66731/
certificateauthorities/
f0430e1736ea8ed30cc5d5de9af67a7e3586bad9ae4d89c2a44163f65fdd8cf6",
            "GroupCertificateAuthorityId":
 "f0430e1736ea8ed30cc5d5de9af67a7e3586bad9ae4d89c2a44163f65fdd8cf6"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListGroupCertificateAuthorities의 섹션을 참조하세요. AWS CLI

### `list-group-versions`

다음 코드 예시에서는 `list-group-versions`을 사용하는 방법을 보여 줍니다.

AWS CLI

Greengrass 그룹의 버전을 나열하려면

다음 `list-group-versions` 예제에서는 지정된 Greengrass 그룹의 버전을 나열합니다.

```
aws greengrass list-group-versions \
    --group-id "1013db12-8b58-45ff-acc7-704248f66731"
```

출력:

```
{
    "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"
        }
    ]
```

```
}
```

- 자세한 API 내용은 명령 참조ListGroupVersions의 섹션을 참조하세요. AWS CLI

### `list-groups`

다음 코드 예시에서는 `list-groups`을 사용하는 방법을 보여 줍니다.

AWS CLI

Greengrass 그룹을 나열하려면

다음 `list-groups` 예제에서는 AWS 계정에 정의된 모든 Greengrass 그룹을 나열합니다.

```
aws greengrass list-groups
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListGroups의 섹션을 참조하세요. AWS CLI

### `list-logger-definition-versions`

다음 코드 예시에서는 `list-logger-definition-versions`을 사용하는 방법을 보여 줍니다.

AWS CLI

로거 정의의 버전 목록을 가져오려면

다음 `list-logger-definition-versions` 예제에서는 지정된 로거 정의의 모든 버전 목록을 가져옵니다.

```
aws greengrass list-logger-definition-versions \
    --logger-definition-id "49eeeb66-f1d3-4e34-86e3-3617262abf23"
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListLoggerDefinitionVersions의 섹션을 참조하세요. AWS CLI

### `list-logger-definitions`

다음 코드 예시에서는 `list-logger-definitions`을 사용하는 방법을 보여 줍니다.

AWS CLI

로거 정의 목록을 가져오려면

다음 `list-logger-definitions` 예제에서는 AWS 계정에 대한 모든 로거 정의를 나열합니다.

```
aws greengrass list-logger-definitions
```

출력:

```
{
    "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"
```

```
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListLoggerDefinitions의 섹션을 참조하세요. AWS CLI

### `list-resource-definition-versions`

다음 코드 예시에서는 `list-resource-definition-versions`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 정의의 버전을 나열하려면

다음 `list-resource-definition-versions` 예제에서는 지정된 Greengrass 리소스의 버전을 나열합니다.

```
aws greengrass list-resource-definition-versions \
    --resource-definition-id "ad8c101d-8109-4b0e-b97d-9cc5802ab658"
```

출력:

```
{
    "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"
        }
    ]
```

```
}
```

- 자세한 API 내용은 명령 참조ListResourceDefinitionVersions의 섹션을 참조하세요. AWS CLI

### `list-resource-definitions`

다음 코드 예시에서는 `list-resource-definitions`을 사용하는 방법을 보여 줍니다.

AWS CLI

정의된 리소스를 나열하려면

다음 `list-resource-definitions` 예제에서는 AWS IoT Greengrass에서 사용할 수 있도록 정의된 리소스를 나열합니다.

```
aws greengrass list-resource-definitions
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListResourceDefinitions의 섹션을 참조하세요. AWS CLI

### `list-subscription-definition-versions`

다음 코드 예시에서는 `list-subscription-definition-versions`을 사용하는 방법을 보여 줍니다.

AWS CLI

구독 정의의 버전을 나열하려면

다음 `list-subscription-definition-versions` 예제에서는 지정된 구독의 모든 버전을 나열합니다. `list-subscription-definitions` 명령을 사용하여 구독 ID를 조회할 수 있습니다.

```
aws greengrass list-subscription-definition-versions \
    --subscription-definition-id "70e49321-83d5-45d2-bc09-81f4917ae152"
```

출력:

```
{
    "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"
```

```
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListSubscriptionDefinitionVersions의 섹션을 참조하세요. AWS CLI

### `list-subscription-definitions`

다음 코드 예시에서는 `list-subscription-definitions`을 사용하는 방법을 보여 줍니다.

AWS CLI

목록 구독 정의를 가져오려면

다음 `list-subscription-definitions` 예제에서는 AWS 계정에 정의된 모든 AWS IoT Greengrass 구독을 나열합니다.

```
aws greengrass list-subscription-definitions
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListSubscriptionDefinitions의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 연결된 태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 지정된 리소스에 연결된 태그와 해당 값을 나열합니다.

```
aws greengrass list-tags-for-resource \
    --resource-arn "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/resources/ad8c101d-8109-4b0e-b97d-9cc5802ab658"
```

출력:

```
{
    "tags": {
        "ResourceSubType": "USB",
        "ResourceType": "Device"
```

```
    }
}
```

자세한 내용은 AWS IoT Greengrass 개발자 안내서의 Greengrass 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### `reset-deployments`

다음 코드 예시에서는 `reset-deployments`을 사용하는 방법을 보여 줍니다.

AWS CLI

Greengrass 그룹에 대한 배포 정보를 정리하려면

다음 `reset-deployments` 예제에서는 지정된 Greengrass 그룹에 대한 배포 정보를 정리합니다. 를 추가하면 코어 디바이스`--force option`가 응답할 때까지 기다리지 않고 배포 정보가 재설정됩니다.

```
aws greengrass reset-deployments \
    --group-id "1402daf9-71cf-4cfe-8be0-d5e80526d0d8" \
    --force
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS IoT Greengrass 개발자 안내서의 배포 재설정을 참조하세요.

- 자세한 API 내용은 명령 참조ResetDeployments의 섹션을 참조하세요. AWS CLI

### `start-bulk-deployment`

다음 코드 예시에서는 `start-bulk-deployment`을 사용하는 방법을 보여 줍니다.

AWS CLI

대량 배포 작업을 시작하려면

다음 `start-bulk-deployment` 예제에서는 S3 버킷에 저장된 파일을 사용하여 배포할 그룹을 지정하는 대량 배포 작업을 시작합니다.

```
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\"}"
```

출력:

```
{
    "BulkDeploymentArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
bulk/deployments/870fb41b-6288-4e0c-bc76-a7ba4b4d3267",
    "BulkDeploymentId": "870fb41b-6288-4e0c-bc76-a7ba4b4d3267"
}
```

자세한 내용은 AWS IoT Greengrass 개발자 안내서의 그룹에 대한 대량 배포 생성을 참조하세요.

- 자세한 API 내용은 명령 참조StartBulkDeployment의 섹션을 참조하세요. AWS CLI

### `stop-bulk-deployment`

다음 코드 예시에서는 `stop-bulk-deployment`을 사용하는 방법을 보여 줍니다.

AWS CLI

대량 배포를 중지하려면

다음 `stop-bulk-deployment` 예제에서는 지정된 대량 배포를 중지합니다. 완료된 대량 배포를 중지하려고 하면 다음과 같은 오류가 발생합니다. `InvalidInputException: Cannot change state of finished execution.`

```
aws greengrass stop-bulk-deployment \
    --bulk-deployment-id "870fb41b-6288-4e0c-bc76-a7ba4b4d3267"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT Greengrass 개발자 안내서의 그룹에 대한 대량 배포 생성을 참조하세요.

- 자세한 API 내용은 명령 참조StopBulkDeployment의 섹션을 참조하세요. AWS CLI

### tag-resource

다음 코드 예시에서는 tag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 태그를 적용하려면

다음 tag-resource 예제에서는 지정된 Greengrass 리소스ResourceSubType에 두 개의 태그 ResourceType 및 를 적용합니다. 이 작업은 새 태그와 값을 추가하거나 기존 태그의 값을 업데이트할 수 있습니다. untag-resource 명령을 사용하여 태그를 제거합니다.

```
aws greengrass tag-resource \
    --resource-arn "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
definition/resources/ad8c101d-8109-4b0e-b97d-9cc5802ab658" \
    --tags "ResourceType=Device,ResourceSubType=USB"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT Greengrass 개발자 안내서의 Greengrass 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### untag-resource

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에서 태그와 해당 값을 제거하려면

다음 untag-resource 예제에서는 키가 지정된 Greengrass 그룹에서 Category인 태그를 제거합니다. 지정된 리소스에 대한 키Category가 없는 경우 오류가 반환되지 않습니다.

```
aws greengrass untag-resource \
    --resource-arn "arn:aws:greengrass:us-west-2:123456789012:/greengrass/
groups/1013db12-8b58-45ff-acc7-704248f66731" \
```

```
    --tag-keys "Category"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT Greengrass 개발자 안내서의 Greengrass 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-connectivity-info

다음 코드 예시에서는 update-connectivity-info을 사용하는 방법을 보여 줍니다.

AWS CLI

Greengrass 코어의 연결 정보를 업데이트하려면

다음 update-connectivity-info 예제에서는 디바이스가 지정된 Greengrass 코어에 연결하는 데 사용할 수 있는 엔드포인트를 변경합니다. 연결 정보는 IP 주소 또는 도메인 이름의 목록으로, 해당 포트 번호 및 선택적 고객 정의 메타데이터를 포함합니다. 로컬 네트워크가 변경될 때 연결 정보를 업데이트해야 할 수 있습니다.

```
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\"}]"
```

출력:

```
{
    "Version": "312de337-59af-4cf9-a278-2a23bd39c300"
}
```

- 자세한 API 내용은 명령 참조UpdateConnectivityInfo의 섹션을 참조하세요. AWS CLI

### update-connector-definition

다음 코드 예시에서는 update-connector-definition을 사용하는 방법을 보여 줍니다.

AWS CLI

커넥터 정의의 이름을 업데이트하려면

다음 update-connector-definition 예제에서는 지정된 커넥터 정의의 이름을 업데이트합니다. 커넥터의 세부 정보를 업데이트하려면 create-connector-definition-version 명령을 사용하여 새 버전을 생성합니다.

```
aws greengrass update-connector-definition \
    --connector-definition-id "55d0052b-0d7d-44d6-b56f-21867215e118" \
    --name "GreengrassConnectors2019"
```

자세한 내용은 AWS IoT Greengrass 개발자 안내서의 커넥터를 사용하여 서비스 및 프로토콜과 통합을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateConnectorDefinition의 섹션을 참조하세요. AWS CLI

### update-core-definition

다음 코드 예시에서는 update-core-definition을 사용하는 방법을 보여 줍니다.

AWS CLI

코어 정의를 업데이트하려면

다음 update-core-definition 예제에서는 지정된 코어 정의의 이름을 변경합니다. 코어 정의의 name 속성만 업데이트할 수 있습니다.

```
aws greengrass update-core-definition \
    --core-definition-id "582efe12-b05a-409e-9a24-a2ba1bcc4a12" \
    --name "MyCoreDevices"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT Greengrass 개발자 안내서의 IoT Greengrass Core 구성을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조UpdateCoreDefinition의 섹션을 참조하세요. AWS CLI

### update-device-definition

다음 코드 예시에서는 update-device-definition을 사용하는 방법을 보여 줍니다.

AWS CLI

디바이스 정의를 업데이트하려면

다음 update-device-definition 예제에서는 지정된 디바이스 정의의 이름을 변경합니다. 디바이스 정의의 name 속성만 업데이트할 수 있습니다.

```
aws greengrass update-device-definition \
    --device-definition-id "f9ba083d-5ad4-4534-9f86-026a45df1ccd" \
    --name "TemperatureSensors"
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조UpdateDeviceDefinition의 섹션을 참조하세요. AWS CLI

### update-function-definition

다음 코드 예시에서는 update-function-definition을 사용하는 방법을 보여 줍니다.

AWS CLI

함수 정의의 이름을 업데이트하려면

다음 update-function-definition 예제에서는 지정된 함수 정의의 이름을 업데이트합니다. 함수의 세부 정보를 업데이트하려면 create-function-definition-version 명령을 사용하여 새 버전을 생성합니다.

```
aws greengrass update-function-definition \
    --function-definition-id "e47952bd-dea9-4e2c-a7e1-37bbe8807f46" \
    --name ObsoleteFunction
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT Greengrass 개발자 안내서의 로컬 Lambda 함수 실행을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateFunctionDefinition의 섹션을 참조하세요. AWS CLI

### update-group-certificate-configuration

다음 코드 예시에서는 update-group-certificate-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

그룹 인증서의 만료를 업데이트하려면

다음 update-group-certificate-configuration 예제에서는 지정된 그룹에 대해 생성된 인증서에 대해 10일의 만료 기간을 설정합니다.

```
aws greengrass update-group-certificate-configuration \
    --group-id "8eaadd72-ce4b-4f15-892a-0cc4f3a343f1" \
    --certificate-expiry-in-milliseconds 864000000
```

출력:

```
{
    "CertificateExpiryInMilliseconds": 864000000,
    "CertificateAuthorityExpiryInMilliseconds": 2524607999000,
    "GroupId": "8eaadd72-ce4b-4f15-892a-0cc4f3a343f1"
}
```

자세한 내용은 AWS IoT Greengrass 개발자 안내서의 IoT Greengrass 보안을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조UpdateGroupCertificateConfiguration의 섹션을 참조하세요. AWS CLI

### update-group

다음 코드 예시에서는 update-group을 사용하는 방법을 보여 줍니다.

AWS CLI

그룹 이름을 업데이트하려면

다음 update-group 예제에서는 지정된 Greengrass 그룹의 이름을 업데이트합니다. 그룹의 세부 정보를 업데이트하려면 create-group-version 명령을 사용하여 새 버전을 생성합니다.

```
aws greengrass update-group \
    --group-id "1402daf9-71cf-4cfe-8be0-d5e80526d0d8" \
    --name TestGroup4of6
```

자세한 내용은 AWS IoT Greengrass 개발자 안내서의 AWS IoT에서 IoT Greengrass 구성을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조UpdateGroup의 섹션을 참조하세요. AWS CLI

### update-logger-definition

다음 코드 예시에서는 update-logger-definition을 사용하는 방법을 보여 줍니다.

AWS CLI

로거 정의를 업데이트하려면

다음 update-logger-definition 예제에서는 지정된 로거 정의의 이름을 변경합니다. 로거 정의의 name 속성만 업데이트할 수 있습니다.

```
aws greengrass update-logger-definition \
    --logger-definition-id "a454b62a-5d56-4ca9-bdc4-8254e1662cb0" \
    --name "LoggingConfigsForSensors"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT Greengrass 개발자 안내서의 IoT Greengrass Logs로 모니터링을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조UpdateLoggerDefinition의 섹션을 참조하세요. AWS CLI

### update-resource-definition

다음 코드 예시에서는 update-resource-definition을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 정의의 이름을 업데이트하려면

다음 update-resource-definition 예제에서는 지정된 리소스 정의의 이름을 업데이트합니다. 리소스에 대한 세부 정보를 변경하려면 create-resource-definition-version 명령을 사용하여 새 버전을 생성합니다.

```
aws greengrass update-resource-definition \
    --resource-definition-id "c8bb9ebc-c3fd-40a4-9c6a-568d75569d38" \
    --name GreengrassConnectorResources
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT Greengrass 개발자 안내서의 Lambda 함수 및 커넥터를 사용한 로컬 리소스 액세스를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateResourceDefinition의 섹션을 참조하세요. AWS CLI

### update-subscription-definition

다음 코드 예시에서는 update-subscription-definition을 사용하는 방법을 보여 줍니다.

AWS CLI

구독 정의의 이름을 업데이트하려면

다음 update-subscription-definition 예제에서는 지정된 구독 정의의 이름을 업데이트합니다. 구독에 대한 세부 정보를 변경하려면 create-subscription-definition-version 명령을 사용하여 새 버전을 생성합니다.

```
aws greengrass update-subscription-definition \
    --subscription-definition-id "fa81bc84-3f59-4377-a84b-5d0134da359b" \
    --name "ObsoleteSubscription"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 가이드의 제목을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateSubscriptionDefinition의 섹션을 참조하세요. AWS CLI

### update-thing-runtime-configuration

다음 코드 예시에서는 update-thing-runtime-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

Greengrass 코어의 런타임 구성에서 원격 측정을 켜려면

다음 update-thing-runtime-configuration 예제에서는 Greengrass 코어의 런타임 구성을 업데이트하여 원격 측정을 활성화합니다.

```
aws greengrass update-thing-runtime-configuration \
    --thing-name SampleGreengrassCore \
    --telemetry-configuration {\"Telemetry\":\"On\"}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT Greengrass 개발자 안내서의 원격 측정 설정 구성을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateThingRuntimeConfiguration의 섹션을 참조하세요. AWS CLI

## AWS IoT Greengrass V2 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS IoT Greengrass V2.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `associate-service-role-to-account`

다음 코드 예시에서는 `associate-service-role-to-account`을 사용하는 방법을 보여 줍니다.

AWS CLI

Greengrass 서비스 역할을 AWS 계정에 연결하려면

다음 `associate-service-role-to-account` 예제에서는 서비스 역할을 AWS 계정의 AWS IoT Greengrass와 연결합니다.

```
aws greengrassv2 associate-service-role-to-account \
    --role-arn arn:aws:iam::123456789012:role/service-role/Greengrass_ServiceRole
```

출력:

```
{
```

```
    "associatedAt": "2022-01-19T19:21:53Z"
}
```

자세한 내용은 AWS IoT V2 Greengrass V2 개발자 안내서의 Greengrass 서비스 역할을 참조하세요.

- 자세한 API 내용은 명령 참조AssociateServiceRoleToAccount의 섹션을 참조하세요. AWS CLI

### batch-associate-client-device-with-core-device

다음 코드 예시에서는 batch-associate-client-device-with-core-device을 사용하는 방법을 보여 줍니다.

AWS CLI

클라이언트 디바이스를 코어 디바이스와 연결하려면

다음 batch-associate-client-device-with-core-device 예제에서는 두 클라이언트 디바이스를 코어 디바이스와 연결합니다.

```
aws greengrassv2 batch-associate-client-device-with-core-device \
  --core-device-thing-name MyGreengrassCore \
  --entries thingName=MyClientDevice1 thingName=MyClientDevice2
```

출력:

```
{
    "errorEntries": []
}
```

자세한 내용은 IoT Greengrass V2 개발자 안내서의 로컬 IoT 디바이스와의 상호 작용을 참조하세요. AWS IoT V2

- 자세한 API 내용은 명령 참조BatchAssociateClientDeviceWithCoreDevice의 섹션을 참조하세요. AWS CLI

### batch-disassociate-client-device-from-core-device

다음 코드 예시에서는 batch-disassociate-client-device-from-core-device을 사용하는 방법을 보여 줍니다.

AWS CLI

코어 디바이스에서 클라이언트 디바이스를 연결 해제하려면

다음 batch-disassociate-client-device-from-core-device 예제에서는 코어 디바이스에서 두 클라이언트 디바이스의 연결을 해제합니다.

```
aws greengrassv2 batch-disassociate-client-device-from-core-device \
  --core-device-thing-name MyGreengrassCore \
  --entries thingName=MyClientDevice1 thingName=MyClientDevice2
```

출력:

```
{
    "errorEntries": []
}
```

자세한 내용은 IoT Greengrass V2 개발자 안내서의 로컬 IoT 디바이스와의 상호 작용을 참조하세요. AWS IoT V2

- 자세한 API 내용은 명령 참조BatchDisassociateClientDeviceFromCoreDevice의 섹션을 참조하세요. AWS CLI

### cancel-deployment

다음 코드 예시에서는 cancel-deployment을 사용하는 방법을 보여 줍니다.

AWS CLI

배포를 취소하려면

다음 cancel-deployment 예제에서는 사물 그룹에 대한 지속적인 배포를 중지합니다.

```
aws greengrassv2 cancel-deployment \
    --deployment-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

출력:

```
{
    "message": "SUCCESS"
```

```
}
```

자세한 내용은 AWS IoT Greengrass V2 개발자 안내서의 배포 취소를 참조하세요.

- 자세한 API 내용은 명령 참조CancelDeployment의 섹션을 참조하세요. AWS CLI

### `create-component-version`

다음 코드 예시에서는 `create-component-version`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 레시피에서 구성 요소 버전을 생성하려면

다음 `create-component-version` 예제에서는 레시피 파일에서 Hello World 구성 요소의 버전을 생성합니다.

```
aws greengrassv2 create-component-version \
    --inline-recipe fileb://com.example.HelloWorld-1.0.0.json
```

`com.example.HelloWorld-1.0.0.json`의 콘텐츠:

```
{
    "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}'"
            }
        }
```

```
    ]
}
```

출력:

```
{
    "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": {}
    }
}
```

자세한 내용은 AWS IoT Greengrass V2 개발자 안내서의 배포할 사용자 지정 구성 요소 생성 및 구성 요소 업로드를 참조하세요.

예제 2: AWS Lambda 함수에서 구성 요소 버전을 생성하려면

다음 `create-component-version` 예제에서는 AWS Lambda 함수에서 Hello World 구성 요소의 버전을 생성합니다.

```
aws greengrassv2 create-component-version \
    --cli-input-json file://lambda-function-component.json
```

`lambda-function-component.json`의 콘텐츠:

```
{
    "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"
                }
            ]
        }
    }
}
```

출력:

```
{
    "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": {}
    }
}
```

자세한 내용은 AWS IoT Greengrass V2 개발자 안내서의 AWS Lambda 함수 실행을 참조하세요.

- 자세한 API 내용은 명령 참조CreateComponentVersion의 섹션을 참조하세요. AWS CLI

### `create-deployment`

다음 코드 예시에서는 `create-deployment`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 배포 생성

다음 `create-deployment` 예제에서는 AWS IoT Greengrass 명령줄 인터페이스를 코어 디바이스에 배포합니다.

```
aws greengrassv2 create-deployment \
    --cli-input-json file://cli-deployment.json
```

`cli-deployment.json`의 콘텐츠:

```
{
    "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": {}
}
```

출력:

```
{
    "deploymentId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

자세한 내용은 AWS IoT Greengrass V2 개발자 안내서의 배포 생성을 참조하세요.

예제 2: 구성 요소 구성을 업데이트하는 배포 생성

다음 `create-deployment` 예제에서는 AWS IoT Greengrass 핵 구성 요소를 코어 디바이스 그룹에 배포합니다. 이 배포는 핵 구성 요소에 대해 다음과 같은 구성 업데이트를 적용합니다.

대상 디바이스의 프록시 설정을 기본 프록시 없음 설정으로 재설정합니다.대상 디바이스의 MQTT 설정을 기본값으로 재설정합니다.핵의 JVM 옵션을 설정합니다JVM.핵의 로깅 수준을 설정합니다.

```
aws greengrassv2 create-deployment \
    --cli-input-json file://nucleus-deployment.json
```

`nucleus-deployment.json`의 콘텐츠:

```
{
    "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": {}
}
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS IoT Greengrass V2 개발자 안내서의 배포 생성 및 구성 요소 구성 업데이트를 참조하세요.

- 자세한 API 내용은 명령 참조CreateDeployment의 섹션을 참조하세요. AWS CLI

### delete-component

다음 코드 예시에서는 delete-component을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 요소 버전을 삭제하려면

다음 delete-component 예제에서는 Hello World 구성 요소를 삭제합니다.

```
aws greengrassv2 delete-component \
    --arn arn:aws:greengrass:us-
west-2:123456789012:components:com.example.HelloWorld:versions:1.0.0
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT Greengrass V2 개발자 안내서의 구성 요소 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteComponent의 섹션을 참조하세요. AWS CLI

### delete-core-device

다음 코드 예시에서는 delete-core-device을 사용하는 방법을 보여 줍니다.

AWS CLI

코어 디바이스를 삭제하려면

다음 delete-core-device 예제에서는 AWS IoT Greengrass 코어 디바이스를 삭제합니다.

```
aws greengrassv2 delete-core-device \
    --core-device-thing-name MyGreengrassCore
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT Greengrass V2 개발자 안내서의 IoT Greengrass Core 소프트웨어 제거를 참조하세요. AWS IoT V2

- 자세한 API 내용은 명령 참조DeleteCoreDevice의 섹션을 참조하세요. AWS CLI

### describe-component

다음 코드 예시에서는 describe-component을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 요소 버전을 설명하려면

다음 `describe-component` 예제에서는 Hello World 구성 요소에 대해 설명합니다.

```
aws greengrassv2 describe-component \
    --arn arn:aws:greengrass:us-
west-2:123456789012:components:com.example.HelloWorld:versions:1.0.0
```

출력:

```
{
    "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"
            }
        }
    ]
}
```

자세한 내용은 AWS IoT Greengrass V2 개발자 안내서의 구성 요소 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeComponent의 섹션을 참조하세요. AWS CLI

### `disassociate-service-role-from-account`

다음 코드 예시에서는 `disassociate-service-role-from-account`을 사용하는 방법을 보여 줍니다.

AWS CLI

Greengrass 서비스 역할을 AWS 계정에서 연결 해제하려면

다음 `disassociate-service-role-from-account` 예제에서는 Greengrass 서비스 역할을 AWS 계정의 AWS IoT Greengrass에서 연결 해제합니다.

```
aws greengrassv2 disassociate-service-role-from-account
```

출력:

```
{
    "disassociatedAt": "2022-01-19T19:26:09Z"
}
```

자세한 내용은 AWS IoT V2 Greengrass V2 개발자 안내서의 Greengrass 서비스 역할을 참조하세요.

- 자세한 API 내용은 명령 참조DisassociateServiceRoleFromAccount의 섹션을 참조하세요. AWS CLI

### `get-component-version-artifact`

다음 코드 예시에서는 `get-component-version-artifact`을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 요소 아티팩트를 다운로드할 URL 를 가져오려면

다음 `get-component-version-artifact` 예제에서는 URL를 가져와 로컬 디버그 콘솔 구성 요소의 JAR 파일을 다운로드합니다.

```
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"
```

출력:

```
{
```

```
    "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..."
}
```

자세한 내용은 AWS IoT Greengrass V2 개발자 안내서의 구성 요소 관리를 참조하세요.

- 자세한 API 내용은 명령 참조GetComponentVersionArtifact의 섹션을 참조하세요. AWS CLI

### `get-component`

다음 코드 예시에서는 `get-component`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 구성 요소의 레시피를 YAML 형식(Linux, macOS 또는 Unix)으로 다운로드하려면

다음 `get-component` 예제에서는 Hello World 구성 요소의 레시피를 YAML 형식으로 파일에 다운로드합니다. 이 명령은 다음 작업을 수행합니다.

`--output` 및 `--query` 파라미터를 사용하여 명령의 출력을 제어합니다. 이러한 파라미터는 명령의 출력에서 레시피 블롭을 추출합니다. 출력 제어에 대한 자세한 내용은 명령줄 인터페이스 사용 설명서의 명령 출력 제어를 참조하세요. `base64` 유틸리티를 사용합니다. AWS  이 유틸리티는 추출된 블롭을 원본 텍스트로 디코딩합니다. 성공한 `get-component` 명령으로 반환되는 Blob은 base64 인코딩 텍스트입니다. 원본 텍스트를 얻으려면 이 블롭을 디코딩해야 합니다.디코딩된 텍스트를 파일에 저장합니다. 명령의 마지막 섹션(`> com.example.HelloWorld-1.0.0.json`)은 디코딩된 텍스트를 파일에 저장합니다.

```
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
```

자세한 내용은 AWS IoT Greengrass V2 개발자 안내서의 구성 요소 관리를 참조하세요.

예제 2: 구성 요소의 레시피를 YAML 형식으로 다운로드하려면(Windows CMD)

다음 get-component 예제에서는 Hello World 구성 요소의 레시피를 YAML 형식으로 파일에 다운로드합니다. 이 명령은 certutil 유틸리티를 사용합니다.

```
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
```

자세한 내용은 AWS IoT Greengrass V2 개발자 안내서의 구성 요소 관리를 참조하세요.

예제 3: 구성 요소의 레시피를 YAML 형식으로 다운로드하려면(Windows PowerShell)

다음 get-component 예제에서는 Hello World 구성 요소의 레시피를 YAML 형식으로 파일에 다운로드합니다. 이 명령은 certutil 유틸리티를 사용합니다.

```
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
```

자세한 내용은 AWS IoT Greengrass V2 개발자 안내서의 구성 요소 관리를 참조하세요.

- 자세한 API 내용은 명령 참조GetComponent의 섹션을 참조하세요. AWS CLI

### get-connectivity-info

다음 코드 예시에서는 get-connectivity-info을 사용하는 방법을 보여 줍니다.

AWS CLI

Greengrass 코어 디바이스의 연결 정보를 가져오려면

다음 get-connectivity-info 예제에서는 Greengrass 코어 디바이스의 연결 정보를 가져옵니다. 클라이언트 디바이스는 이 정보를 사용하여 이 코어 디바이스에서 실행되는 MQTT브로커에 연결합니다.

```
aws greengrassv2 get-connectivity-info \
    --thing-name MyGreengrassCore
```

출력:

```
{
    "connectivityInfo": [
        {
            "id": "localIP_192.0.2.0",
            "hostAddress": "192.0.2.0",
            "portNumber": 8883
        }
    ]
}
```

자세한 내용은 AWS IoT Greengrass V2 개발자 안내서의 코어 디바이스 엔드포인트 관리를 참조하세요.

- 자세한 API 내용은 명령 참조GetConnectivityInfo의 섹션을 참조하세요. AWS CLI

### get-core-device

다음 코드 예시에서는 get-core-device을 사용하는 방법을 보여 줍니다.

AWS CLI

코어 디바이스를 가져오려면

다음 get-core-device 예제에서는 AWS IoT Greengrass 코어 디바이스에 대한 정보를 가져옵니다.

```
aws greengrassv2 get-core-device \
    --core-device-thing-name MyGreengrassCore
```

출력:

```
{
```

```
    "coreDeviceThingName": "MyGreengrassCore",
    "coreVersion": "2.0.3",
    "platform": "linux",
    "architecture": "amd64",
    "status": "HEALTHY",
    "lastStatusUpdateTimestamp": "2021-01-08T04:57:58.838000-08:00",
    "tags": {}
}
```

자세한 내용은 AWS IoT Greengrass V2 개발자 안내서의 코어 디바이스 상태 확인을 참조하세요.

- 자세한 API 내용은 명령 참조GetCoreDevice의 섹션을 참조하세요. AWS CLI

### get-deployment

다음 코드 예시에서는 get-deployment을 사용하는 방법을 보여 줍니다.

AWS CLI

배포를 가져오려면

다음 get-deployment 예제에서는 AWS IoT Greengrass 핵 구성 요소를 코어 디바이스 그룹에 배포하는 방법에 대한 정보를 가져옵니다.

```
aws greengrassv2 get-deployment \
    --deployment-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

출력:

```
{
    "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": {}
}
```

자세한 내용은 AWS IoT Greengrass V2 개발자 안내서의 디바이스에 구성 요소 배포를 참조하세요.

- 자세한 API 내용은 명령 참조GetDeployment의 섹션을 참조하세요. AWS CLI

### `get-service-role-for-account`

다음 코드 예시에서는 `get-service-role-for-account`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 계정의 Greengrass 서비스 역할을 가져오려면

다음 `get-service-role-for-account` 예제에서는 AWS 계정의 AWS IoT Greengrass와 연결된 서비스 역할을 가져옵니다.

```
aws greengrassv2 get-service-role-for-account
```

출력:

```
{
    "associatedAt": "2022-01-19T19:21:53Z",
    "roleArn": "arn:aws:iam::123456789012:role/service-role/Greengrass_ServiceRole"
}
```

자세한 내용은 AWS IoT V2 Greengrass V2 개발자 안내서의 Greengrass 서비스 역할을 참조하세요.

- 자세한 API 내용은 명령 참조GetServiceRoleForAccount의 섹션을 참조하세요. AWS CLI

### `list-client-devices-associated-with-core-device`

다음 코드 예시에서는 `list-client-devices-associated-with-core-device`을 사용하는 방법을 보여 줍니다.

AWS CLI

코어 디바이스와 연결된 클라이언트 디바이스를 나열하려면

다음 `list-client-devices-associated-with-core-device` 예제에서는 코어 디바이스와 연결된 모든 클라이언트 디바이스를 나열합니다.

```
aws greengrassv2 list-client-devices-associated-with-core-device \
  --core-device-thing-name MyTestGreengrassCore
```

출력:

```
{
    "associatedClientDevices": [
        {
            "thingName": "MyClientDevice2",
            "associationTimestamp": "2021-07-12T16:33:55.843000-07:00"
        },
        {
            "thingName": "MyClientDevice1",
            "associationTimestamp": "2021-07-12T16:33:55.843000-07:00"
        }
    ]
}
```

자세한 내용은 IoT Greengrass V2 개발자 안내서의 로컬 IoT 디바이스와의 상호 작용을 참조하세요. AWS IoT V2

- 자세한 API 내용은 명령 참조ListClientDevicesAssociatedWithCoreDevice의 섹션을 참조하세요. AWS CLI

### `list-component-versions`

다음 코드 예시에서는 `list-component-versions`을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 요소의 버전을 나열하려면

다음 `list-component-versions` 예제에서는 Hello World 구성 요소의 모든 버전을 나열합니다.

```
aws greengrassv2 list-component-versions \
    --arn arn:aws:greengrass:us-
west-2:123456789012:components:com.example.HelloWorld
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS IoT Greengrass V2 개발자 안내서의 구성 요소 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ListComponentVersions의 섹션을 참조하세요. AWS CLI

### `list-components`

다음 코드 예시에서는 `list-components`을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 요소를 나열하려면

다음 `list-components` 예제에서는 현재 리전의 AWS 계정에 정의된 각 구성 요소와 최신 버전을 나열합니다.

```
aws greengrassv2 list-components
```

출력:

```
{
    "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"
                        }
                    }
                ]
            }
        }
    ]
}
```

자세한 내용은 AWS IoT Greengrass V2 개발자 안내서의 구성 요소 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ListComponents의 섹션을 참조하세요. AWS CLI

### list-core-devices

다음 코드 예시에서는 list-core-devices을 사용하는 방법을 보여 줍니다.

AWS CLI

코어 디바이스를 나열하려면

다음 list-core-devices 예제에서는 현재 리전의 AWS 계정에 있는 AWS IoT Greengrass 코어 디바이스를 나열합니다.

```
aws greengrassv2 list-core-devices
```

출력:

```
{
    "coreDevices": [
        {
            "coreDeviceThingName": "MyGreengrassCore",
            "status": "HEALTHY",
            "lastStatusUpdateTimestamp": "2021-01-08T04:57:58.838000-08:00"
        }
    ]
}
```

자세한 내용은 AWS IoT Greengrass V2 개발자 안내서의 코어 디바이스 상태 확인을 참조하세요.

- 자세한 API 내용은 명령 참조ListCoreDevices의 섹션을 참조하세요. AWS CLI

### list-deployments

다음 코드 예시에서는 list-deployments을 사용하는 방법을 보여 줍니다.

AWS CLI

배포를 나열하려면

다음 list-deployments 예제에서는 현재 리전의 AWS 계정에 정의된 각 배포의 최신 개정을 나열합니다.

```
aws greengrassv2 list-deployments
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 AWS IoT Greengrass V2 개발자 안내서의 디바이스에 구성 요소 배포를 참조하세요.

- 자세한 API 내용은 명령 참조ListDeployments의 섹션을 참조하세요. AWS CLI

### `list-effective-deployments`

다음 코드 예시에서는 `list-effective-deployments`을 사용하는 방법을 보여 줍니다.

AWS CLI

배포 작업을 나열하려면

다음 `list-effective-deployments` 예제에서는 AWS IoT Greengrass 코어 디바이스에 적용되는 배포를 나열합니다.

```
aws greengrassv2 list-effective-deployments \
```

```
    --core-device-thing-name MyGreengrassCore
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS IoT Greengrass V2 개발자 안내서의 코어 디바이스 상태 확인을 참조하세요.

- 자세한 API 내용은 명령 참조ListEffectiveDeployments의 섹션을 참조하세요. AWS CLI

### `list-installed-components`

다음 코드 예시에서는 `list-installed-components`을 사용하는 방법을 보여 줍니다.

AWS CLI

코어 디바이스에 설치된 구성 요소를 나열하려면

다음 `list-installed-components` 예제에서는 AWS IoT Greengrass 코어 디바이스에 설치된 구성 요소를 나열합니다.

```
aws greengrassv2 list-installed-components \
    --core-device-thing-name MyGreengrassCore
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 AWS IoT Greengrass V2 개발자 안내서의 코어 디바이스 상태 확인을 참조하세요.

- 자세한 API 내용은 명령 참조ListInstalledComponents의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스의 태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 AWS IoT Greengrass 코어 디바이스의 모든 태그를 나열합니다.

```
aws greengrassv2 list-tags-for-resource \
    --resource-arn arn:aws:greengrass:us-
west-2:123456789012:coreDevices:MyGreengrassCore
```

출력:

```
{
    "tags": {
        "Owner": "richard-roe"
    }
}
```

자세한 내용은 AWS IoT Greengrass V2 개발자 안내서의 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### `tag-resource`

다음 코드 예시에서는 `tag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 태그를 추가하려면

다음 `tag-resource` 예제에서는 AWS IoT Greengrass 코어 디바이스에 소유자 태그를 추가합니다. 이 태그를 사용하여 코어 디바이스를 소유한 사람에 따라 코어 디바이스에 대한 액세스를 제어할 수 있습니다.

```
aws greengrassv2 tag-resource \
    --resource-arn arn:aws:greengrass:us-
west-2:123456789012:coreDevices:MyGreengrassCore \
    --tags Owner=richard-roe
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT Greengrass V2 개발자 안내서의 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### `untag-resource`

다음 코드 예시에서는 `untag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에서 태그를 제거하려면

다음 `untag-resource` 예제에서는 AWS IoT Greengrass 코어 디바이스에서 소유자 태그를 제거합니다.

```
aws iotsitewise untag-resource \
    --resource-arn arn:aws:greengrass:us-
west-2:123456789012:coreDevices:MyGreengrassCore \
    --tag-keys Owner
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT Greengrass V2 개발자 안내서의 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-connectivity-info

다음 코드 예시에서는 `update-connectivity-info`을 사용하는 방법을 보여 줍니다.

AWS CLI

Greengrass 코어 디바이스의 연결 정보를 업데이트하려면

다음 `update-connectivity-info` 예제에서는 Greengrass 코어 디바이스의 연결 정보를 가져옵니다. 클라이언트 디바이스는 이 정보를 사용하여 이 코어 디바이스에서 실행되는 MQTT브로커에 연결합니다.

```
aws greengrassv2 update-connectivity-info \
    --thing-name MyGreengrassCore \
    --cli-input-json file://core-device-connectivity-info.json
```

`core-device-connectivity-info.json`의 콘텐츠:

```
{
    "connectivityInfo": [
        {
            "hostAddress": "192.0.2.0",
            "portNumber": 8883,
            "id": "localIP_192.0.2.0"
        }
```

```
    ]
}
```

출력:

```
{
    "version": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

자세한 내용은 AWS IoT Greengrass V2 개발자 안내서의 코어 디바이스 엔드포인트 관리를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateConnectivityInfo의 섹션을 참조하세요. AWS CLI

## AWS IoT Jobs SDK release 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS IoT Jobs SDK release.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `describe-job-execution`

다음 코드 예시에서는 `describe-job-execution`을 사용하는 방법을 보여 줍니다.

AWS CLI

작업 실행의 세부 정보를 가져오려면

다음 `describe-job-execution` 예제에서는 지정된 작업 및 사물의 최신 실행 세부 정보를 검색합니다.

```
aws iot-jobs-data describe-job-execution \
    --job-id SampleJob \
    --thing-name MotionSensor1 \
    --endpoint-url https://1234567890abcd.jobs.iot.us-west-2.amazonaws.com
```

출력:

```
{
    "execution": {
        "approximateSecondsBeforeTimedOut": 88,
        "executionNumber": 2939653338,
        "jobId": "SampleJob",
        "lastUpdatedAt": 1567701875.743,
        "queuedAt": 1567701902.444,
        "status": "QUEUED",
        "thingName": "MotionSensor1 ",
        "versionNumber": 3
   }
}
```

자세한 내용은 AWS IoT 개발자 안내서의 디바이스 및 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeJobExecution의 섹션을 참조하세요. AWS CLI

### get-pending-job-executions

다음 코드 예시에서는 get-pending-job-executions을 사용하는 방법을 보여 줍니다.

AWS CLI

사물의 터미널 상태가 아닌 모든 작업 목록을 가져오려면

다음 get-pending-job-executions 예제에서는 지정된 사물에 대해 터미널 상태가 아닌 모든 작업의 목록을 표시합니다.

```
aws iot-jobs-data get-pending-job-executions \
    --thing-name MotionSensor1
    --endpoint-url https://1234567890abcd.jobs.iot.us-west-2.amazonaws.com
```

출력:

```
{
```

```
    "inProgressJobs": [
    ],
    "queuedJobs": [
        {
            "executionNumber": 2939653338,
            "jobId": "SampleJob",
            "lastUpdatedAt": 1567701875.743,
            "queuedAt": 1567701902.444,
            "versionNumber": 3
      }
    ]
}
```

자세한 내용은 AWS IoT 개발자 안내서의 디바이스 및 작업을 참조하세요.

- 자세한 API 내용은 명령 참조GetPendingJobExecutions의 섹션을 참조하세요. AWS CLI

### `start-next-pending-job-execution`

다음 코드 예시에서는 `start-next-pending-job-execution`을 사용하는 방법을 보여 줍니다.

AWS CLI

사물에 대해 보류 중인 다음 작업 실행을 가져오고 시작하려면

다음 `start-next-pending-job-execution` 예제에서는 지정된 사물의 상태가 IN_PROGRESS 또는 인 다음 작업 실행QUEUED을 검색하고 시작합니다.

```
aws iot-jobs-data start-next-pending-job-execution \
    --thing-name MotionSensor1
    --endpoint-url https://1234567890abcd.jobs.iot.us-west-2.amazonaws.com
```

출력:

```
{
    "execution": {
        "approximateSecondsBeforeTimedOut": 88,
        "executionNumber": 2939653338,
        "jobId": "SampleJob",
        "lastUpdatedAt": 1567714853.743,
        "queuedAt": 1567701902.444,
        "startedAt": 1567714871.690,
        "status": "IN_PROGRESS",
```

```
        "thingName": "MotionSensor1 ",
        "versionNumber": 3
    }
}
```

자세한 내용은 AWS IoT 개발자 안내서의 디바이스 및 작업을 참조하세요.

- 자세한 API 내용은 명령 참조StartNextPendingJobExecution의 섹션을 참조하세요. AWS CLI

**update-job-execution**

다음 코드 예시에서는 update-job-execution을 사용하는 방법을 보여 줍니다.

AWS CLI

작업 실행 상태를 업데이트하려면

다음 update-job-execution 예제에서는 지정된 작업 및 사물의 상태를 업데이트합니다.

```
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
```

출력:

```
{
    "executionState": {
        "status": "REMOVED",
        "versionNumber": 3
    },
}
```

자세한 내용은 AWS IoT 개발자 안내서의 디바이스 및 작업을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateJobExecution의 섹션을 참조하세요. AWS CLI

## AWS IoT SiteWise 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS IoT SiteWise.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `associate-assets`

다음 코드 예시에서는 `associate-assets`을 사용하는 방법을 보여 줍니다.

AWS CLI

하위 자산을 상위 자산에 연결하려면

다음 `associate-assets` 예제는 풍력 터빈 자산을 풍력 팜 자산에 연결합니다. 풍력 터빈 자산 모델은 풍력 팜 자산 모델의 계층 구조로 존재합니다.

```
aws iotsitewise associate-assets \
    --asset-id a1b2c3d4-5678-90ab-cdef-44444EXAMPLE \
    --hierarchy-id a1b2c3d4-5678-90ab-cdef-77777EXAMPLE \
    --child-asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT SiteWise 사용 설명서의 자산 연결을 참조하세요.

- 자세한 API 내용은 명령 참조AssociateAssets의 섹션을 참조하세요. AWS CLI

### `batch-associate-project-assets`

다음 코드 예시에서는 `batch-associate-project-assets`을 사용하는 방법을 보여 줍니다.

AWS CLI

자산을 프로젝트에 연결하려면

다음 batch-associate-project-assets 예제에서는 풍력 팜 자산을 프로젝트에 연결합니다.

```
aws iotsitewise batch-associate-project-assets \
    --project-id a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE \
    --asset-ids a1b2c3d4-5678-90ab-cdef-44444EXAMPLE
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT SiteWise Monitor Application Guide의 프로젝트에 자산 추가를 참조하세요.

- 자세한 API 내용은 명령 참조BatchAssociateProjectAssets의 섹션을 참조하세요. AWS CLI

### batch-disassociate-project-assets

다음 코드 예시에서는 batch-disassociate-project-assets을 사용하는 방법을 보여 줍니다.

AWS CLI

프로젝트에서 자산을 연결 해제하려면

다음 batch-disassociate-project-assets 예제에서는 풍력 팜 자산을 프로젝트에서 연결 해제합니다.

```
aws iotsitewise batch-disassociate-project-assets \
    --project-id a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE \
    --asset-ids a1b2c3d4-5678-90ab-cdef-44444EXAMPLE
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT SiteWise Monitor 애플리케이션 안내서의 프로젝트에 자산 추가를 참조하세요.

- 자세한 API 내용은 명령 참조BatchDisassociateProjectAssets의 섹션을 참조하세요. AWS CLI

### batch-put-asset-property-value

다음 코드 예시에서는 batch-put-asset-property-value을 사용하는 방법을 보여 줍니다.

AWS CLI

자산 속성으로 데이터를 보내려면

다음 batch-put-asset-property-value 예제에서는 속성 별칭으로 식별된 자산 속성으로 전력 및 온도 데이터를 전송합니다.

```
aws iotsitewise batch-put-asset-property-value \
    --cli-input-json file://batch-put-asset-property-value.json
```

batch-put-asset-property-value.json의 콘텐츠:

```
{
    "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
                    }
                }
            ]
        }
    ]
}
```

출력:

```
{
    "errorEntries": []
}
```

자세한 내용은 AWS IoT SiteWise API 사용 설명서의 IoT를 사용하여 데이터 수집을 참조하세요. AWS IoT SiteWise

- 자세한 API 내용은 명령 참조BatchPutAssetPropertyValue의 섹션을 참조하세요. AWS CLI

### `create-access-policy`

다음 코드 예시에서는 `create-access-policy`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 포털에 대한 사용자 관리 액세스 권한 부여

다음 `create-access-policy` 예제에서는 풍력 발전 단지 회사의 웹 포털에 대한 사용자 관리 액세스 권한을 부여하는 액세스 정책을 생성합니다.

```
aws iotsitewise create-access-policy \
    --cli-input-json file://create-portal-administrator-access-policy.json
```

`create-portal-administrator-access-policy.json`의 콘텐츠:

```
{
    "accessPolicyIdentity": {
        "user": {
            "id": "a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbEXAMPLE"
        }
    },
    "accessPolicyPermission": "ADMINISTRATOR",
    "accessPolicyResource": {
        "portal": {
            "id": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE"
        }
    }
}
```

출력:

```
{
    "accessPolicyId": "a1b2c3d4-5678-90ab-cdef-cccccEXAMPLE",
    "accessPolicyArn": "arn:aws:iotsitewise:us-west-2:123456789012:access-policy/
a1b2c3d4-5678-90ab-cdef-cccccEXAMPLE"
}
```

자세한 내용은 AWS IoT SiteWise 사용 설명서의 포털 관리자 추가 또는 제거를 참조하세요.

예제 2: 사용자에게 프로젝트에 대한 읽기 전용 액세스 권한을 부여하려면

다음 `create-access-policy` 예제에서는 사용자에게 풍력 발전 단지 프로젝트에 대한 읽기 전용 액세스 권한을 부여하는 액세스 정책을 생성합니다.

```
aws iotsitewise create-access-policy \
    --cli-input-json file://create-project-viewer-access-policy.json
```

`create-project-viewer-access-policy.json`의 콘텐츠:

```
{
    "accessPolicyIdentity": {
        "user": {
            "id": "a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbEXAMPLE"
        }
    },
    "accessPolicyPermission": "VIEWER",
    "accessPolicyResource": {
        "project": {
            "id": "a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE"
        }
    }
}
```

출력:

```
{
    "accessPolicyId": "a1b2c3d4-5678-90ab-cdef-dddddEXAMPLE",
    "accessPolicyArn": "arn:aws:iotsitewise:us-west-2:123456789012:access-policy/
a1b2c3d4-5678-90ab-cdef-dddddEXAMPLE"
}
```

자세한 내용은 AWS IoT SiteWise Monitor Application Guide의 프로젝트 뷰어 할당을 참조하세요.

- 자세한 API 내용은 명령 참조CreateAccessPolicy의 섹션을 참조하세요. AWS CLI

### `create-asset-model`

다음 코드 예시에서는 `create-asset-model`을 사용하는 방법을 보여 줍니다.

AWS CLI

자산 모델을 생성하려면

다음 `create-asset-model` 예제에서는 다음과 같은 속성을 가진 풍력 터빈을 정의하는 자산 모델을 생성합니다.

일련 번호 - 풍력의 일련 번호 turbineGenerated - 풍력 turbineTemperature C에서 생성된 전력 데이터 스트림 - CelsiusTemperature F의 풍력 터빈에서 온도 데이터 스트림 - 섭씨에서 화씨로 매핑된 온도 데이터 포인트

```
aws iotsitewise create-asset-model \
    --cli-input-json file://create-wind-turbine-model.json
```

`create-wind-turbine-model.json`의 콘텐츠:

```
{
    "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"
                        }
```

```
                    }
                }
            }
        }
    ]
}
```

출력:

```
{
    "assetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
    "assetModelArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
    "assetModelStatus": {
        "state": "CREATING"
    }
}
```

자세한 내용은 AWS IoT SiteWise 사용 설명서의 자산 모델 정의를 참조하세요.

- 자세한 API 내용은 명령 참조CreateAssetModel의 섹션을 참조하세요. AWS CLI

### `create-asset`

다음 코드 예시에서는 `create-asset`을 사용하는 방법을 보여 줍니다.

AWS CLI

자산을 생성하려면

다음 `create-asset` 예제에서는 풍력 터빈 자산 모델에서 풍력 터빈 자산을 생성합니다.

```
aws iotsitewise create-asset \
    --asset-model-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE \
    --asset-name "Wind Turbine 1"
```

출력:

```
{
    "assetId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
    "assetArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset/
a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
    "assetStatus": {
```

```
        "state": "CREATING"
    }
}
```

자세한 내용은 AWS IoT SiteWise 사용 설명서의 자산 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateAsset의 섹션을 참조하세요. AWS CLI

### create-dashboard

다음 코드 예시에서는 create-dashboard을 사용하는 방법을 보여 줍니다.

AWS CLI

대시보드를 생성하려면

다음 create-dashboard 예제에서는 풍력 발전 단지에 대해 생성된 총 전력을 표시하는 선 차트가 있는 대시보드를 생성합니다.

```
aws iotsitewise create-dashboard \
    --project-id a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE \
    --dashboard-name "Wind Farm" \
    --dashboard-definition file://create-wind-farm-dashboard.json
```

create-wind-farm-dashboard.json의 콘텐츠:

```
{
    "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"
                }
            ]
```

```
        }
    ]
}
```

출력:

```
{
    "dashboardId": "a1b2c3d4-5678-90ab-cdef-fffffEXAMPLE",
    "dashboardArn": "arn:aws:iotsitewise:us-west-2:123456789012:dashboard/
a1b2c3d4-5678-90ab-cdef-fffffEXAMPLE"
}
```

자세한 내용은 AWS IoT SiteWise 사용 설명서의 대시보드 생성(CLI)을 참조하세요.

- 자세한 API 내용은 명령 참조CreateDashboard의 섹션을 참조하세요. AWS CLI

### `create-gateway`

다음 코드 예시에서는 `create-gateway`을 사용하는 방법을 보여 줍니다.

AWS CLI

게이트웨이를 생성하려면

다음 `create-gateway` 예제에서는 AWS IoT Greengrass에서 실행되는 게이트웨이를 생성합니다.

```
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}
```

출력:

```
{
    "gatewayId": "a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",
    "gatewayArn": "arn:aws:iotsitewise:us-west-2:123456789012:gateway/
a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE"
}
```

자세한 내용은 AWS IoT SiteWise 사용 설명서의 게이트웨이 구성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateGateway의 섹션을 참조하세요. AWS CLI

### create-portal

다음 코드 예시에서는 create-portal을 사용하는 방법을 보여 줍니다.

AWS CLI

포털을 생성하려면

다음 create-portal 예제에서는 풍력 발전 단지 회사를 위한 웹 포털을 생성합니다. AWS Single Sign-On을 활성화한 동일한 리전에서만 포털을 생성할 수 있습니다.

```
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
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS IoT SiteWise 사용 설명서의 IoT 모니터 시작하기 및AWS IoT SiteWise 사용 설명서의 활성화 AWS SSO를 참조하세요. AWS IoT SiteWise

- 자세한 API 내용은 명령 참조CreatePortal의 섹션을 참조하세요. AWS CLI

### create-project

다음 코드 예시에서는 create-project을 사용하는 방법을 보여 줍니다.

AWS CLI

프로젝트를 생성하려면

다음 `create-project` 예제에서는 풍력 발전 단지 프로젝트를 생성합니다.

```
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."
```

출력:

```
{
    "projectId": "a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE",
    "projectArn": "arn:aws:iotsitewise:us-west-2:123456789012:project/
a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE"
}
```

자세한 내용은 AWS IoT SiteWise Monitor 애플리케이션 안내서의 프로젝트 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateProject의 섹션을 참조하세요. AWS CLI

## delete-access-policy

다음 코드 예시에서는 `delete-access-policy`을 사용하는 방법을 보여 줍니다.

AWS CLI

프로젝트 또는 포털에 대한 사용자의 액세스를 취소하려면

다음 `delete-access-policy` 예제에서는 포털에 대한 사용자 관리 액세스 권한을 부여하는 액세스 정책을 삭제합니다.

```
aws iotsitewise delete-access-policy \
    --access-policy-id a1b2c3d4-5678-90ab-cdef-cccccEXAMPLE
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT SiteWise 사용 설명서의 포털 관리자 추가 또는 제거를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteAccessPolicy의 섹션을 참조하세요. AWS CLI

### delete-asset-model

다음 코드 예시에서는 delete-asset-model을 사용하는 방법을 보여 줍니다.

AWS CLI

자산 모델을 삭제하려면

다음 delete-asset-model 예제에서는 풍력 터빈 자산 모델을 삭제합니다.

```
aws iotsitewise delete-asset-model \
    --asset-model-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

출력:

```
{
    "assetModelStatus": {
        "state": "DELETING"
    }
}
```

자세한 내용은 AWS IoT SiteWise 사용 설명서의 자산 모델 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteAssetModel의 섹션을 참조하세요. AWS CLI

### delete-asset

다음 코드 예시에서는 delete-asset을 사용하는 방법을 보여 줍니다.

AWS CLI

자산을 삭제하려면

다음 delete-asset 예제에서는 풍력 터빈 자산을 삭제합니다.

```
aws iotsitewise delete-asset \
    --asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE
```

출력:

```
{
    "assetStatus": {
        "state": "DELETING"
```

```
    }
}
```

자세한 내용은 AWS IoT SiteWise 사용 설명서의 자산 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteAsset의 섹션을 참조하세요. AWS CLI

### `delete-dashboard`

다음 코드 예시에서는 `delete-dashboard`을 사용하는 방법을 보여 줍니다.

AWS CLI

대시보드를 삭제하려면

다음 `delete-dashboard` 예제에서는 풍력 터빈 대시보드를 삭제합니다.

```
aws iotsitewise delete-dashboard \
    --dashboard-id a1b2c3d4-5678-90ab-cdef-fffffEXAMPLE
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT SiteWise Monitor 애플리케이션 안내서의 대시보드 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteDashboard의 섹션을 참조하세요. AWS CLI

### `delete-gateway`

다음 코드 예시에서는 `delete-gateway`을 사용하는 방법을 보여 줍니다.

AWS CLI

게이트웨이를 삭제하려면

다음 `delete-gateway` 예제에서는 게이트웨이를 삭제합니다.

```
aws iotsitewise delete-gateway \
    --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT SiteWise 사용 설명서의 게이트웨이를 사용하여 데이터 수집을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteGateway의 섹션을 참조하세요. AWS CLI

### delete-portal

다음 코드 예시에서는 delete-portal을 사용하는 방법을 보여 줍니다.

AWS CLI

포털을 삭제하려면

다음 delete-portal 예제에서는 풍력 발전 단지 회사의 웹 포털을 삭제합니다.

```
aws iotsitewise delete-portal \
    --portal-id a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE
```

출력:

```
{
    "portalStatus": {
        "state": "DELETING"
    }
}
```

자세한 내용은 AWS IoT SiteWise 사용 설명서의 포털 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeletePortal의 섹션을 참조하세요. AWS CLI

### delete-project

다음 코드 예시에서는 delete-project을 사용하는 방법을 보여 줍니다.

AWS CLI

프로젝트를 삭제하려면

다음 delete-project 예제에서는 풍력 팜 프로젝트를 삭제합니다.

```
aws iotsitewise delete-project \
    --project-id a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT SiteWise Monitor 애플리케이션 안내서의 프로젝트 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteProject의 섹션을 참조하세요. AWS CLI

### describe-access-policy

다음 코드 예시에서는 describe-access-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

액세스 정책을 설명하려면

다음 describe-access-policy 예제에서는 풍력 발전 단지 회사의 웹 포털에 대한 사용자 관리 액세스 권한을 부여하는 액세스 정책을 설명합니다.

```
aws iotsitewise describe-access-policy \
    --access-policy-id a1b2c3d4-5678-90ab-cdef-cccccEXAMPLE
```

출력:

```
{
    "accessPolicyId": "a1b2c3d4-5678-90ab-cdef-cccccEXAMPLE",
    "accessPolicyArn": "arn:aws:iotsitewise:us-west-2:123456789012:access-policy/
a1b2c3d4-5678-90ab-cdef-cccccEXAMPLE",
    "accessPolicyIdentity": {
        "user": {
            "id": "a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbEXAMPLE"
        }
    },
    "accessPolicyResource": {
        "portal": {
            "id": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE"
        }
    },
    "accessPolicyPermission": "ADMINISTRATOR",
    "accessPolicyCreationDate": "2020-02-20T22:35:15.552880124Z",
    "accessPolicyLastUpdateDate": "2020-02-20T22:35:15.552880124Z"
}
```

자세한 내용은 AWS IoT SiteWise 사용 설명서의 포털 관리자 추가 또는 제거를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeAccessPolicy의 섹션을 참조하세요. AWS CLI

### describe-asset-model

다음 코드 예시에서는 describe-asset-model을 사용하는 방법을 보여 줍니다.

AWS CLI

자산 모델을 설명하려면

다음 describe-asset-model 예제에서는 풍력 팜 자산 모델을 설명합니다.

```
aws iotsitewise describe-asset-model \
    --asset-model-id a1b2c3d4-5678-90ab-cdef-22222EXAMPLE
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS IoT SiteWise 사용 설명서의 특정 자산 모델 설명을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeAssetModel의 섹션을 참조하세요. AWS CLI

### `describe-asset-property`

다음 코드 예시에서는 `describe-asset-property`을 사용하는 방법을 보여 줍니다.

AWS CLI

자산 속성을 설명하려면

다음 describe-asset-property 예제에서는 풍력 발전 단지 자산의 총 생성 전력 속성을 설명합니다.

```
aws iotsitewise describe-asset-property \
    --asset-id a1b2c3d4-5678-90ab-cdef-44444EXAMPLE \
    --property-id a1b2c3d4-5678-90ab-cdef-99999EXAMPLE
```

출력:

```
{
    "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"
                    }
                }
            }
        }
```

```
    }
}
```

자세한 내용은 AWS IoT SiteWise 사용 설명서의 특정 자산 속성 설명을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeAssetProperty의 섹션을 참조하세요. AWS CLI

### describe-asset

다음 코드 예시에서는 describe-asset을 사용하는 방법을 보여 줍니다.

AWS CLI

자산을 설명하려면

다음 describe-asset 예제에서는 풍력 팜 자산을 설명합니다.

```
aws iotsitewise describe-asset \
    --asset-id a1b2c3d4-5678-90ab-cdef-44444EXAMPLE
```

출력:

```
{
    "assetId": "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE",
    "assetArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset/
a1b2c3d4-5678-90ab-cdef-44444EXAMPLE",
    "assetName": "Wind Farm 1",
    "assetModelId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
    "assetProperties": [
        {
            "id": "a1b2c3d4-5678-90ab-cdef-88888EXAMPLE",
            "name": "Region",
            "dataType": "STRING"
        },
        {
            "id": "a1b2c3d4-5678-90ab-cdef-99999EXAMPLE",
            "name": "Total Generated Power",
            "dataType": "DOUBLE",
            "unit": "kW"
        }
    ],
    "assetHierarchies": [
        {
```

```
            "id": "a1b2c3d4-5678-90ab-cdef-77777EXAMPLE",
            "name": "Wind Turbines"
        }
    ],
    "assetCreationDate": 1575672453.0,
    "assetLastUpdateDate": 1575672453.0,
    "assetStatus": {
        "state": "ACTIVE"
    }
}
```

자세한 내용은 AWS IoT SiteWise 사용 설명서의 특정 자산 설명을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeAsset의 섹션을 참조하세요. AWS CLI

### describe-dashboard

다음 코드 예시에서는 describe-dashboard을 사용하는 방법을 보여 줍니다.

AWS CLI

대시보드를 설명하려면

다음 describe-dashboard 예제에서는 지정된 풍력 기지 대시보드를 설명합니다.

```
aws iotsitewise describe-dashboard \
    --dashboard-id a1b2c3d4-5678-90ab-cdef-fffffEXAMPLE
```

출력:

```
{
    "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-eeeeeEXAMPLE",
    "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"
}
```

자세한 내용은 AWS IoT SiteWise Monitor 애플리케이션 안내서의 대시보드 보기를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeDashboard의 섹션을 참조하세요. AWS CLI

### describe-gateway-capability-configuration

다음 코드 예시에서는 describe-gateway-capability-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

게이트웨이 기능을 설명하려면

다음 describe-gateway-capability-configuration 예제에서는 OPC-UA 소스 기능에 대해 설명합니다.

```
aws iotsitewise describe-gateway-capability-configuration \
    --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE \
    --capability-namespace "iotsitewise:opcuacollector:1"
```

출력:

```
{
    "gatewayId": "a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",
    "capabilityNamespace": "iotsitewise:opcuacollector:1",
    "capabilityConfiguration": "{\"sources\":[{\"name\":\"Wind Farm #1\",
\"endpoint\":{\"certificateTrust\":{\"type\":\"TrustAny\"},\"endpointUri
\":\"opc.tcp://203.0.113.0:49320\",\"securityPolicy\":\"BASIC256\",
\"messageSecurityMode\":\"SIGN_AND_ENCRYPT\",\"identityProvider\":
{\"type\":\"Username\",\"usernameSecretArn\":\"arn:aws:secretsmanager:us-
east-1:123456789012:secret:greengrass-factory1-auth-3QNDmM\"},\"nodeFilterRules\":
[]},\"measurementDataStreamPrefix\":\"\"}]}",
    "capabilitySyncStatus": "IN_SYNC"
}
```

자세한 내용은 AWS IoT SiteWise 사용 설명서의 데이터 소스 구성을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeGatewayCapabilityConfiguration의 섹션을 참조하세요. AWS CLI

### describe-gateway

다음 코드 예시에서는 describe-gateway을 사용하는 방법을 보여 줍니다.

AWS CLI

게이트웨이를 설명하려면

다음 describe-gateway 예제에서는 게이트웨이에 대해 설명합니다.

```
aws iotsitewise describe-gateway \
    --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE
```

출력:

```
{
    "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
}
```

자세한 내용은 AWS IoT SiteWise 사용 설명서의 게이트웨이를 사용하여 데이터 수집을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeGateway의 섹션을 참조하세요. AWS CLI

### describe-logging-options

다음 코드 예시에서는 describe-logging-options을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 AWS IoT SiteWise 로깅 옵션을 검색하려면

다음 describe-logging-options 예제에서는 현재 리전의 AWS 계정에 대한 현재 AWS IoT SiteWise 로깅 옵션을 검색합니다.

```
aws iotsitewise describe-logging-options
```

출력:

```
{
    "loggingOptions": {
        "level": "INFO"
    }
}
```

자세한 내용은 AWS IoT SiteWise 사용 설명서의 Amazon CloudWatch Logs를 사용한 IoT 모니터링을 참조하세요. AWS IoT SiteWise

- 자세한 API 내용은 명령 참조DescribeLoggingOptions의 섹션을 참조하세요. AWS CLI

### describe-portal

다음 코드 예시에서는 describe-portal을 사용하는 방법을 보여 줍니다.

AWS CLI

포털을 설명하려면

다음 describe-portal 예제에서는 풍력 발전 단지 회사의 웹 포털에 대해 설명합니다.

```
aws iotsitewise describe-portal \
    --portal-id a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE
```

출력:

```
{
```

```
    "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"
}
```

자세한 내용은 AWS IoT SiteWise 사용 설명서의 포털 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DescribePortal의 섹션을 참조하세요. AWS CLI

### `describe-project`

다음 코드 예시에서는 `describe-project`을 사용하는 방법을 보여 줍니다.

AWS CLI

프로젝트를 설명하려면

다음 `describe-project` 예제에서는 풍력 발전 단지 프로젝트를 설명합니다.

```
aws iotsitewise describe-project \
    --project-id a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE
```

출력:

```
{
    "projectId": "a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE",
    "projectArn": "arn:aws:iotsitewise:us-west-2:123456789012:project/
a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE",
    "projectName": "Wind Farm 1",
    "portalId": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
```

```
    "projectDescription": "Contains asset visualizations for Wind Farm #1 for
 Example Corp.",
    "projectCreationDate": "2020-02-20T21:58:43.362246001Z",
    "projectLastUpdateDate": "2020-02-20T21:58:43.362246095Z"
}
```

자세한 내용은 AWS IoT SiteWise Monitor 애플리케이션 안내서의 프로젝트 세부 정보 보기를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeProject의 섹션을 참조하세요. AWS CLI

### `disassociate-assets`

다음 코드 예시에서는 `disassociate-assets`을 사용하는 방법을 보여 줍니다.

AWS CLI

상위 자산에서 하위 자산을 연결 해제하려면

다음 `disassociate-assets` 예제에서는 풍력 터빈 자산을 풍력 팜 자산과 연결 해제합니다.

```
aws iotsitewise disassociate-assets \
    --asset-id a1b2c3d4-5678-90ab-cdef-44444EXAMPLE \
    --hierarchy-id a1b2c3d4-5678-90ab-cdef-77777EXAMPLE \
    --child-asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT SiteWise 사용 설명서의 자산 연결을 참조하세요.

- 자세한 API 내용은 명령 참조DisassociateAssets의 섹션을 참조하세요. AWS CLI

### `get-asset-property-aggregates`

다음 코드 예시에서는 `get-asset-property-aggregates`을 사용하는 방법을 보여 줍니다.

AWS CLI

자산 속성의 집계된 평균 및 개수 값을 검색하려면

다음 `get-asset-property-aggregates` 예제에서는 1시간 동안 풍력 터빈 자산의 평균 총 전력과 총 전력 데이터 포인트 수를 검색합니다.

```
aws iotsitewise get-asset-property-aggregates \
    --asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE \
    --property-id a1b2c3d4-5678-90ab-cdef-66666EXAMPLE \
    --start-date 1580849400 \
    --end-date 1580853000 \
    --aggregate-types AVERAGE COUNT \
    --resolution 1h
```

출력:

```
{
    "aggregatedValues": [
        {
            "timestamp": 1580850000.0,
            "quality": "GOOD",
            "value": {
                "average": 8723.46538886233,
                "count": 12.0
            }
        }
    ]
}
```

자세한 내용은 AWS IoT SiteWise 사용 설명서의 자산 속성 집계 쿼리를 참조하세요.

- 자세한 API 내용은 명령 참조GetAssetPropertyAggregates의 섹션을 참조하세요. AWS CLI

### get-asset-property-value-history

다음 코드 예시에서는 get-asset-property-value-history을 사용하는 방법을 보여 줍니다.

AWS CLI

자산 속성의 기록 값을 검색하려면

다음 get-asset-property-value-history 예제에서는 20분 동안 풍력 터빈 자산의 총 전력 값을 검색합니다.

```
aws iotsitewise get-asset-property-value-history \
    --asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE \
    --property-id a1b2c3d4-5678-90ab-cdef-66666EXAMPLE \
    --start-date 1580851800 \
```

```
    --end-date 1580853000
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS IoT SiteWise 사용 설명서의 과거 자산 속성 값 쿼리를 참조하세요.

- 자세한 API 내용은 명령 참조GetAssetPropertyValueHistory의 섹션을 참조하세요. AWS CLI

### get-asset-property-value

다음 코드 예시에서는 get-asset-property-value을 사용하는 방법을 보여 줍니다.

AWS CLI

자산 속성의 현재 값을 검색하려면

다음 get-asset-property-value 예제에서는 풍력 터빈 자산의 현재 총 전력을 검색합니다.

```
aws iotsitewise get-asset-property-value \
    --asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE \
    --property-id a1b2c3d4-5678-90ab-cdef-66666EXAMPLE
```

출력:

```
{
    "propertyValue": {
        "value": {
            "doubleValue": 6890.8677520453875
        },
        "timestamp": {
            "timeInSeconds": 1580853000,
            "offsetInNanos": 0
        },
        "quality": "GOOD"
    }
}
```

자세한 내용은 AWS IoT SiteWise 사용 설명서의 현재 자산 속성 값 쿼리를 참조하세요.

- 자세한 API 내용은 명령 참조GetAssetPropertyValue의 섹션을 참조하세요. AWS CLI

### list-access-policies

다음 코드 예시에서는 list-access-policies을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 액세스 정책을 나열하려면

다음 list-access-policies 예제에서는 포털 관리자인 사용자의 모든 액세스 정책을 나열합니다.

```
aws iotsitewise list-access-policies \
    --identity-type USER \
    --identity-id a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbEXAMPLE
```

출력:

```
{
    "accessPolicySummaries": [
        {
            "id": "a1b2c3d4-5678-90ab-cdef-cccccEXAMPLE",
            "identity": {
                "user": {
                    "id": "a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbEXAMPLE"
                }
            },
            "resource": {
                "portal": {
                    "id": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE"
                }
            },
            "permission": "ADMINISTRATOR"
        }
    ]
}
```

자세한 내용은 AWS IoT SiteWise 사용 설명서의 포털 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ListAccessPolicies의 섹션을 참조하세요. AWS CLI

### list-asset-models

다음 코드 예시에서는 list-asset-models을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 자산 모델을 나열하려면

다음 `list-asset-models` 예제에서는 현재 리전의 AWS 계정에 정의된 모든 자산 모델을 나열합니다.

```
aws iotsitewise list-asset-models
```

출력:

```
{
    "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"
            }
        }
    ]
}
```

자세한 내용은 AWS IoT SiteWise 사용 설명서의 모든 자산 모델 나열을 참조하세요.

- 자세한 API 내용은 명령 참조ListAssetModels의 섹션을 참조하세요. AWS CLI

### `list-assets`

다음 코드 예시에서는 `list-assets`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 모든 최상위 자산을 나열하려면

다음 `list-assets` 예제에서는 자산 계층 구조 트리에서 최상위 수준이고 현재 리전의 AWS 계정에 정의된 모든 자산을 나열합니다.

```
aws iotsitewise list-assets \
    --filter TOP_LEVEL
```

출력:

```
{
    "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"
                }
            ]
        }
    ]
}
```

자세한 내용은 AWS IoT SiteWise 사용 설명서의 자산 나열을 참조하세요.

예제 2: 자산 모델을 기반으로 모든 자산을 나열하려면

다음 `list-assets` 예제에서는 자산 모델을 기반으로 현재 리전의 AWS 계정에 정의된 모든 자산을 나열합니다.

```
aws iotsitewise list-assets \
    --asset-model-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

출력:

```
{
    "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": []
        }
    ]
}
```

자세한 내용은 AWS IoT SiteWise 사용 설명서의 자산 나열을 참조하세요.

- 자세한 API 내용은 명령 참조ListAssets의 섹션을 참조하세요. AWS CLI

### `list-associated-assets`

다음 코드 예시에서는 `list-associated-assets`을 사용하는 방법을 보여 줍니다.

AWS CLI

특정 계층 구조의 자산에 연결된 모든 자산을 나열하려면

다음 `list-associated-assets` 예제에서는 지정된 풍력 기지 자산과 연결된 모든 풍력 터빈 자산을 나열합니다.

```
aws iotsitewise list-associated-assets \
    --asset-id a1b2c3d4-5678-90ab-cdef-44444EXAMPLE \
    --hierarchy-id a1b2c3d4-5678-90ab-cdef-77777EXAMPLE
```

출력:

```
{
    "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": []
        }
    ]
}
```

자세한 내용은 AWS IoT SiteWise 사용 설명서의 특정 자산과 연결된 자산 나열을 참조하세요.

- 자세한 API 내용은 명령 참조ListAssociatedAssets의 섹션을 참조하세요. AWS CLI

### `list-dashboards`

다음 코드 예시에서는 `list-dashboards`을 사용하는 방법을 보여 줍니다.

AWS CLI

프로젝트의 모든 대시보드를 나열하려면

다음 `list-dashboards` 예제에서는 프로젝트에 정의된 모든 대시보드를 나열합니다.

```
aws iotsitewise list-dashboards \
    --project-id a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE
```

출력:

```
{
    "dashboardSummaries": [
        {
            "id": "a1b2c3d4-5678-90ab-cdef-fffffEXAMPLE",
            "name": "Wind Farm",
            "creationDate": "2020-05-01T20:32:12.228476348Z",
            "lastUpdateDate": "2020-05-01T20:32:12.228476348Z"
        }
    ]
}
```

자세한 내용은 AWS IoT SiteWise Monitor 애플리케이션 안내서의 대시보드 보기를 참조하세요.

- 자세한 API 내용은 명령 참조ListDashboards의 섹션을 참조하세요. AWS CLI

### `list-gateways`

다음 코드 예시에서는 `list-gateways`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 게이트웨이를 나열하려면

다음 `list-gateways` 예제에서는 현재 리전의 AWS 계정에 정의된 모든 게이트웨이를 나열합니다.

```
aws iotsitewise list-gateways
```

출력:

```
{
    "gatewaySummaries": [
        {
            "gatewayId": "a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",
            "gatewayName": "ExampleCorpGateway",
            "gatewayCapabilitySummaries": [
                {
                    "capabilityNamespace": "iotsitewise:opcuacollector:1",
                    "capabilitySyncStatus": "IN_SYNC"
                }
            ],
            "creationDate": 1588369971.457,
```

```
            "lastUpdateDate": 1588369971.457
        }
    ]
}
```

자세한 내용은 AWS IoT SiteWise 사용 설명서의 게이트웨이를 사용하여 데이터 수집을 참조하세요.

- 자세한 API 내용은 명령 참조ListGateways의 섹션을 참조하세요. AWS CLI

### `list-portals`

다음 코드 예시에서는 `list-portals`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 포털을 나열하려면

다음 `list-portals` 예제에서는 현재 리전의 AWS 계정에 정의된 모든 포털을 나열합니다.

```
aws iotsitewise list-portals
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS IoT SiteWise 사용 설명서의 포털 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ListPortals의 섹션을 참조하세요. AWS CLI

### list-project-assets

다음 코드 예시에서는 list-project-assets을 사용하는 방법을 보여 줍니다.

AWS CLI

프로젝트에 연결된 모든 자산을 나열하려면

다음 list-project-assets 예제에서는 풍력 발전 단지 프로젝트와 연결된 모든 자산을 나열합니다.

```
aws iotsitewise list-projects \
    --project-id a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE
```

출력:

```
{
    "assetIds": [
        "a1b2c3d4-5678-90ab-cdef-44444EXAMPLE"
    ]
}
```

자세한 내용은 AWS IoT SiteWise Monitor 애플리케이션 안내서의 프로젝트에 자산 추가를 참조하세요.

- 자세한 API 내용은 명령 참조ListProjectAssets의 섹션을 참조하세요. AWS CLI

### list-projects

다음 코드 예시에서는 list-projects을 사용하는 방법을 보여 줍니다.

AWS CLI

포털의 모든 프로젝트를 나열하려면

다음 list-projects 예제에서는 포털에 정의된 모든 프로젝트를 나열합니다.

```
aws iotsitewise list-projects \
```

```
    --portal-id a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE
```

출력:

```
{
    "projectSummaries": [
        {
            "id": "a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE",
            "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"
        }
    ]
}
```

자세한 내용은 AWS IoT SiteWise Monitor 애플리케이션 안내서의 프로젝트 세부 정보 보기를 참조하세요.

- 자세한 API 내용은 명령 참조ListProjects의 섹션을 참조하세요. AWS CLI

### list-tags-for-resource

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스의 모든 태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 풍력 터빈 자산의 모든 태그를 나열합니다.

```
aws iotsitewise list-tags-for-resource \
    --resource-arn arn:aws:iotsitewise:us-west-2:123456789012:asset/
a1b2c3d4-5678-90ab-cdef-33333EXAMPLE
```

출력:

```
{
    "tags": {
        "Owner": "richard-roe"
```

```
    }
}
```

자세한 내용은 AWS IoT SiteWise 사용 설명서의 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### put-logging-options

다음 코드 예시에서는 put-logging-options을 사용하는 방법을 보여 줍니다.

AWS CLI

로깅 수준을 지정하려면

다음 put-logging-options 예제에서는 AWS IoT SiteWise 에서 INFO 수준 로깅을 활성화합니다. 다른 수준에는 DEBUG 및 가 포함됩니다OFF.

```
aws iotsitewise put-logging-options \
    --logging-options level=INFO
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT SiteWise 사용 설명서의 Amazon CloudWatch Logs를 사용한 IoT 모니터링을 참조하세요. AWS IoT SiteWise

- 자세한 API 내용은 명령 참조PutLoggingOptions의 섹션을 참조하세요. AWS CLI

### tag-resource

다음 코드 예시에서는 tag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 태그를 추가하려면

다음 tag-resource 예제에서는 풍력 터빈 자산에 소유자 태그를 추가합니다. 이를 통해 자산 소유자를 기반으로 자산에 대한 액세스를 제어할 수 있습니다.

```
aws iotsitewise tag-resource \
```

```
    --resource-arn arn:aws:iotsitewise:us-west-2:123456789012:asset/
a1b2c3d4-5678-90ab-cdef-33333EXAMPLE \
    --tags Owner=richard-roe
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT SiteWise 사용 설명서의 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### untag-resource

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에서 태그를 제거하려면

다음 untag-resource 예제에서는 풍력 터빈 자산에서 소유자 태그를 제거합니다.

```
aws iotsitewise untag-resource \
    --resource-arn arn:aws:iotsitewise:us-west-2:123456789012:asset/
a1b2c3d4-5678-90ab-cdef-33333EXAMPLE \
    --tag-keys Owner
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT SiteWise 사용 설명서의 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-access-policy

다음 코드 예시에서는 update-access-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

프로젝트 시청자에게 프로젝트의 소유권을 부여하려면

다음 update-access-policy 예제에서는 프로젝트 뷰어에게 프로젝트의 소유권을 부여하는 액세스 정책을 업데이트합니다.

```
aws iotsitewise update-access-policy \
    --access-policy-id a1b2c3d4-5678-90ab-cdef-dddddEXAMPLE \
    --cli-input-json file://update-project-viewer-access-policy.json
```

update-project-viewer-access-policy.json의 콘텐츠:

```
{
    "accessPolicyIdentity": {
        "user": {
            "id": "a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbEXAMPLE"
        }
    },
    "accessPolicyPermission": "ADMINISTRATOR",
    "accessPolicyResource": {
        "project": {
            "id": "a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE"
        }
    }
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT SiteWise Monitor 애플리케이션 안내서의 프로젝트 소유자 할당을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateAccessPolicy의 섹션을 참조하세요. AWS CLI

### update-asset-model

다음 코드 예시에서는 update-asset-model을 사용하는 방법을 보여 줍니다.

AWS CLI

자산 모델을 업데이트하려면

다음 update-asset-model 예제에서는 풍력 팜 자산 모델의 설명을 업데이트합니다. 이 예제에서는 가 기존 모델을 새 모델로 update-asset-model 덮어쓰기 때문에 모델의 기존 IDs 및 정의를 포함합니다.

```
aws iotsitewise update-asset-model \
    --cli-input-json file://update-wind-farm-model.json
```

update-wind-farm-model.json의 콘텐츠:

```
{
    "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"
        }
    ]
}
```

출력:

```
{
    "assetModelId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
    "assetModelArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/
a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
    "assetModelStatus": {
        "state": "CREATING"
    }
}
```

자세한 내용은 AWS IoT SiteWise 사용 설명서의 자산 모델 업데이트를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateAssetModel의 섹션을 참조하세요. AWS CLI

### update-asset-property

다음 코드 예시에서는 update-asset-property을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 자산 속성의 별칭을 업데이트하려면

다음 update-asset-property 예제에서는 풍력 터빈 자산의 전력 속성 별칭을 업데이트합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT SiteWise 사용 설명서의 자산 속성에 산업 데이터 스트림 매핑을 참조하세요.

예제 2: 자산 속성 알림을 활성화하려면

다음 update-asset-property 예제에서는 풍력 터빈 자산의 전력 속성에 대한 자산 속성 업데이트 알림을 활성화합니다. 속성 값 업데이트는 MQTT 주제 에 게시되며$aws/sitewise/asset-models/<assetModelId>/assets/<assetId>/properties/<propertyId>, 여기서 각 ID는 자산 속성의 속성, 자산 및 모델 ID로 대체됩니다.

```
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"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT SiteWise 사용 설명서의 다른 서비스와 상호 작용을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateAssetProperty의 섹션을 참조하세요. AWS CLI

### update-asset

다음 코드 예시에서는 update-asset을 사용하는 방법을 보여 줍니다.

AWS CLI

자산 이름을 업데이트하려면

다음 update-asset 예제에서는 풍력 터빈 자산의 이름을 업데이트합니다.

```
aws iotsitewise update-asset \
    --asset-id a1b2c3d4-5678-90ab-cdef-33333EXAMPLE \
    --asset-name "Wind Turbine 2"
```

출력:

```
{
    "assetStatus": {
        "state": "UPDATING"
    }
}
```

자세한 내용은 AWS IoT SiteWise 사용 설명서의 자산 업데이트를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateAsset의 섹션을 참조하세요. AWS CLI

### update-dashboard

다음 코드 예시에서는 update-dashboard을 사용하는 방법을 보여 줍니다.

AWS CLI

대시보드를 업데이트하려면

다음 update-dashboard 예제에서는 풍력 발전 단지에 대해 생성된 총 전력을 표시하는 대시보드의 선 차트 제목을 변경합니다.

```
aws iotsitewise update-dashboard \
    --project-id a1b2c3d4-5678-90ab-cdef-fffffEXAMPLE \
    --dashboard-name "Wind Farm" \
    --dashboard-definition file://update-wind-farm-dashboard.json
```

update-wind-farm-dashboard.json의 콘텐츠:

```
{
    "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"
                }
            ]
        }
    ]
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT SiteWise 사용 설명서의 대시보드 생성(CLI)을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateDashboard의 섹션을 참조하세요. AWS CLI

### update-gateway-capability-configuration

다음 코드 예시에서는 update-gateway-capability-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

게이트웨이 기능을 업데이트하려면

다음 update-gateway-capability-configuration 예제에서는 다음 속성을 사용하여 OPC-UA 소스를 구성합니다.

모든 인증서를 신뢰합니다. Basic256 알고리즘을 사용하여 메시지를 보호합니다. SignAndEncrypt 모드를 사용하여 연결을 보호합니다. AWS Secrets Manager 보안 암호에 저장된 인증 자격 증명을 사용합니다.

```
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
```

opc-ua-capability-configuration.json의 콘텐츠:

```
{
    "sources": [
        {
            "name": "Wind Farm #1",
            "endpoint": {
                "certificateTrust": {
                    "type": "TrustAny"
                },
                "endpointUri": "opc.tcp://203.0.113.0:49320",
                "securityPolicy": "BASIC256",
                "messageSecurityMode": "SIGN_AND_ENCRYPT",
                "identityProvider": {
                    "type": "Username",
```

```
                    "usernameSecretArn": "arn:aws:secretsmanager:us-
west-2:123456789012:secret:greengrass-windfarm1-auth-1ABCDE"
                },
                "nodeFilterRules": []
            },
            "measurementDataStreamPrefix": ""
        }
    ]
}
```

출력:

```
{
    "capabilityNamespace": "iotsitewise:opcuacollector:1",
    "capabilitySyncStatus": "OUT_OF_SYNC"
}
```

자세한 내용은 AWS IoT SiteWise 사용 설명서의 데이터 소스 구성을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateGatewayCapabilityConfiguration의 섹션을 참조하세요. AWS CLI

### update-gateway

다음 코드 예시에서는 update-gateway을 사용하는 방법을 보여 줍니다.

AWS CLI

게이트웨이 이름을 업데이트하려면

다음 update-gateway 예제에서는 게이트웨이의 이름을 업데이트합니다.

```
aws iotsitewise update-gateway \
    --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE \
    --gateway-name ExampleCorpGateway1
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT SiteWise 사용 설명서의 게이트웨이를 사용하여 데이터 수집을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateGateway의 섹션을 참조하세요. AWS CLI

### update-portal

다음 코드 예시에서는 update-portal을 사용하는 방법을 보여 줍니다.

AWS CLI

포털의 세부 정보를 업데이트하려면

다음 update-portal 예제에서는 풍력 발전 단지 회사의 웹 포털을 업데이트합니다.

```
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
```

출력:

```
{
    "portalStatus": {
        "state": "UPDATING"
    }
}
```

자세한 내용은 AWS IoT SiteWise 사용 설명서의 포털 관리를 참조하세요.

- 자세한 API 내용은 명령 참조UpdatePortal의 섹션을 참조하세요. AWS CLI

### update-project

다음 코드 예시에서는 update-project을 사용하는 방법을 보여 줍니다.

AWS CLI

프로젝트 세부 정보를 업데이트하려면

다음 update-project 예제에서는 풍력 팜 프로젝트를 업데이트합니다.

```
aws iotsitewise update-project \
    --project-id a1b2c3d4-5678-90ab-cdef-eeeeeEXAMPLE \
    --project-name "Wind Farm 1" \
```

```
    --project-description "Contains asset visualizations for Wind Farm #1 for
Example Corp."
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT SiteWise Monitor 애플리케이션 안내서의 프로젝트 세부 정보 변경을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateProject의 섹션을 참조하세요. AWS CLI

## AWS IoT Things Graph 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS IoT Things Graph.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `associate-entity-to-thing`

다음 코드 예시에서는 `associate-entity-to-thing`을 사용하는 방법을 보여 줍니다.

AWS CLI

사물을 디바이스와 연결하려면

다음 `associate-entity-to-thing` 예제에서는 사물을 디바이스와 연결합니다. 이 예제에서는 퍼블릭 네임스페이스에 있는 모션 센서 디바이스를 사용합니다.

```
aws iotthingsgraph associate-entity-to-thing \
    --thing-name "MotionSensorName" \
    --entity-id "urn:tdm:aws/examples:Device:HCSR501MotionSensor"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 사물 그래프 사용 설명서의 모델 생성 및 업로드를 참조하세요.

- 자세한 API 내용은 명령 참조AssociateEntityToThing의 섹션을 참조하세요. AWS CLI

### create-flow-template

다음 코드 예시에서는 create-flow-template을 사용하는 방법을 보여 줍니다.

AWS CLI

흐름을 생성하려면

다음 create-flow-template 예제에서는 흐름(워크플로)을 생성합니다. 의 값은 흐름을 모델링하는 GraphQLMyFlowDefinition입니다.

```
aws iotthingsgraph create-flow-template \
    --definition language=GRAPHQL,text="MyFlowDefinition"
```

출력:

```
{
    "summary": {
        "createdAt": 1559248067.545,
        "id": "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow",
        "revisionNumber": 1
    }
}
```

자세한 내용은 AWS IoT 사물 그래프 사용 설명서의 흐름 작업을 참조하세요.

- 자세한 API 내용은 명령 참조CreateFlowTemplate의 섹션을 참조하세요. AWS CLI

### create-system-instance

다음 코드 예시에서는 create-system-instance을 사용하는 방법을 보여 줍니다.

AWS CLI

시스템 인스턴스를 생성하려면

다음 create-system-instance 예제에서는 시스템 인스턴스를 생성합니다. 의 값은 시스템 인스턴스를 모델링하는 GraphQLMySystemInstanceDefinition입니다.

```
aws iotthingsgraph create-system-instance -\
    -definition language=GRAPHQL,text="MySystemInstanceDefinition" \
    --target CLOUD \
    --flow-actions-role-arn myRoleARN
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 AWS IoT 사물 그래프 사용 설명서의 시스템 및 흐름 구성 작업을 참조하세요.

- 자세한 API 내용은 명령 참조CreateSystemInstance의 섹션을 참조하세요. AWS CLI

### create-system-template

다음 코드 예시에서는 create-system-template을 사용하는 방법을 보여 줍니다.

AWS CLI

시스템을 생성하려면

다음 create-system-template 예제에서는 시스템을 생성합니다. 의 값은 시스템을 모델링하는 GraphQL MySystemDefinition 입니다.

```
aws iotthingsgraph create-system-template \
    --definition language=GRAPHQL,text="MySystemDefinition"
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 AWS IoT 사물 그래프 사용 설명서의 시스템 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateSystemTemplate의 섹션을 참조하세요. AWS CLI

### delete-flow-template

다음 코드 예시에서는 delete-flow-template을 사용하는 방법을 보여 줍니다.

AWS CLI

흐름을 삭제하려면

다음 delete-flow-template 예제에서는 흐름(워크플로)을 삭제합니다.

```
aws iotthingsgraph delete-flow-template \
    --id "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 사물 그래프 사용 설명서의 IoT 사물 그래프 개체, 흐름, 시스템 및 배포의 수명 주기 관리를 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조DeleteFlowTemplate의 섹션을 참조하세요. AWS CLI

### delete-namespace

다음 코드 예시에서는 delete-namespace을 사용하는 방법을 보여 줍니다.

AWS CLI

네임스페이스를 삭제하려면

다음 delete-namespace 예제에서는 네임스페이스를 삭제합니다.

```
aws iotthingsgraph delete-namespace
```

출력:

```
{
   "namespaceArn": "arn:aws:iotthingsgraph:us-west-2:123456789012",
   "namespaceName": "us-west-2/123456789012/default"
}
```

자세한 내용은 AWS IoT 사물 그래프 사용 설명서의 IoT 사물 그래프 개체, 흐름, 시스템 및 배포의 수명 주기 관리를 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조DeleteNamespace의 섹션을 참조하세요. AWS CLI

### delete-system-instance

다음 코드 예시에서는 delete-system-instance을 사용하는 방법을 보여 줍니다.

AWS CLI

시스템 인스턴스를 삭제하려면

다음 delete-system-instance 예제에서는 시스템 인스턴스를 삭제합니다.

```
aws iotthingsgraph delete-system-instance \
    --id "urn:tdm:us-west-2/123456789012/default:Deployment:Room218"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 사물 그래프 사용 설명서의 IoT 사물 그래프 개체, 흐름, 시스템 및 배포의 수명 주기 관리를 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조DeleteSystemInstance의 섹션을 참조하세요. AWS CLI

### delete-system-template

다음 코드 예시에서는 delete-system-template을 사용하는 방법을 보여 줍니다.

AWS CLI

시스템을 삭제하려면

다음 delete-system-template 예제에서는 시스템을 삭제합니다.

```
aws iotthingsgraph delete-system-template \
    --id "urn:tdm:us-west-2/123456789012/default:System:MySystem"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 사물 그래프 사용 설명서의 IoT 사물 그래프 개체, 흐름, 시스템 및 배포의 수명 주기 관리를 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조DeleteSystemTemplate의 섹션을 참조하세요. AWS CLI

### deploy-system-instance

다음 코드 예시에서는 deploy-system-instance을 사용하는 방법을 보여 줍니다.

AWS CLI

시스템 인스턴스를 배포하려면

다음 delete-system-template 예제에서는 시스템 인스턴스를 배포합니다.

```
aws iotthingsgraph deploy-system-instance \
    --id "urn:tdm:us-west-2/123456789012/default:Deployment:Room218"
```

출력:

```
{
   "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
   }
}
```

자세한 내용은 AWS IoT 사물 그래프 사용 설명서의 시스템 및 흐름 구성 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DeploySystemInstance의 섹션을 참조하세요. AWS CLI

### deprecate-flow-template

다음 코드 예시에서는 deprecate-flow-template을 사용하는 방법을 보여 줍니다.

AWS CLI

흐름 사용을 중단하려면

다음 deprecate-flow-template 예제에서는 흐름(워크플로)을 사용 중지합니다.

```
aws iotthingsgraph deprecate-flow-template \
    --id "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 사물 그래프 사용 설명서의 IoT 사물 그래프 개체, 흐름, 시스템 및 배포의 수명 주기 관리를 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조DeprecateFlowTemplate의 섹션을 참조하세요. AWS CLI

### deprecate-system-template

다음 코드 예시에서는 deprecate-system-template을 사용하는 방법을 보여 줍니다.

AWS CLI

시스템 사용을 중단하려면

다음 deprecate-system-template 예제에서는 시스템의 사용을 중단합니다.

```
aws iotthingsgraph deprecate-system-template \
    --id "urn:tdm:us-west-2/123456789012/default:System:MySystem"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 사물 그래프 사용 설명서의 IoT 사물 그래프 개체, 흐름, 시스템 및 배포의 수명 주기 관리를 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조DeprecateSystemTemplate의 섹션을 참조하세요. AWS CLI

### describe-namespace

다음 코드 예시에서는 describe-namespace을 사용하는 방법을 보여 줍니다.

AWS CLI

네임스페이스에 대한 설명을 가져오려면

다음 `describe-namespace` 예제에서는 네임스페이스에 대한 설명을 가져옵니다.

```
aws iotthingsgraph describe-namespace
```

출력:

```
{
    "namespaceName": "us-west-2/123456789012/default",
    "trackingNamespaceName": "aws",
    "trackingNamespaceVersion": 1,
    "namespaceVersion": 5
}
```

자세한 내용은 AWS IoT 사물 그래프 사용 설명서의 네임스페이스를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeNamespace의 섹션을 참조하세요. AWS CLI

### `dissociate-entity-from-thing`

다음 코드 예시에서는 `dissociate-entity-from-thing`을 사용하는 방법을 보여 줍니다.

AWS CLI

디바이스에서 사물을 연결 해제하려면

다음 `dissociate-entity-from-thing` 예제에서는 디바이스에서 사물을 분리합니다.

```
aws iotthingsgraph dissociate-entity-from-thing \
    --thing-name "MotionSensorName" \
    --entity-type "DEVICE"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 사물 그래프 사용 설명서의 모델 생성 및 업로드를 참조하세요.

- 자세한 API 내용은 명령 참조DissociateEntityFromThing의 섹션을 참조하세요. AWS CLI

### get-entities

다음 코드 예시에서는 get-entities을 사용하는 방법을 보여 줍니다.

AWS CLI

엔터티에 대한 정의를 가져오려면

다음 get-entities 예제에서는 디바이스 모델에 대한 정의를 가져옵니다.

```
aws iotthingsgraph get-entities \
    --ids "urn:tdm:aws/examples:DeviceModel:MotionSensor"
```

출력:

```
{
    "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##
\ntype MotionSensor @deviceModel(id: \"urn:tdm:aws/examples:deviceModel:MotionSensor
\",\n        capability: \"urn:tdm:aws/examples:capability:MotionSensorCapability\")
 {ignore:void}"
            }
        }
    ]
}
```

자세한 내용은 AWS IoT 사물 그래프 사용 설명서의 모델 생성 및 업로드를 참조하세요.

- 자세한 API 내용은 명령 참조GetEntities의 섹션을 참조하세요. AWS CLI

### get-flow-template-revisions

다음 코드 예시에서는 get-flow-template-revisions을 사용하는 방법을 보여 줍니다.

AWS CLI

흐름에 대한 개정 정보를 가져오려면

다음 get-flow-template-revisions 예제에서는 흐름(워크플로)에 대한 개정 정보를 가져옵니다.

```
aws iotthingsgraph get-flow-template-revisions \
    --id urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow
```

출력:

```
{
    "summaries": [
        {
            "id": "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow",
            "revisionNumber": 1,
            "createdAt": 1559247540.292
        }
    ]
}
```

자세한 내용은 AWS IoT 사물 그래프 사용 설명서의 흐름 작업을 참조하세요.

- 자세한 API 내용은 명령 참조GetFlowTemplateRevisions의 섹션을 참조하세요. AWS CLI

### get-flow-template

다음 코드 예시에서는 get-flow-template을 사용하는 방법을 보여 줍니다.

AWS CLI

흐름 정의를 가져오려면

다음 get-flow-template 예제에서는 흐름(워크플로)에 대한 정의를 가져옵니다.

```
aws iotthingsgraph get-flow-template \
    --id "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow"
```

출력:

```
{
    "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}"
        },
        "validatedNamespaceVersion": 5
    }
}
```

자세한 내용은 AWS IoT 사물 그래프 사용 설명서의 흐름 작업을 참조하세요.

- 자세한 API 내용은 명령 참조GetFlowTemplate의 섹션을 참조하세요. AWS CLI

### get-namespace-deletion-status

다음 코드 예시에서는 get-namespace-deletion-status을 사용하는 방법을 보여 줍니다.

AWS CLI

네임스페이스 삭제 작업의 상태를 가져오려면

다음 get-namespace-deletion-status 예제에서는 네임스페이스 삭제 작업의 상태를 가져옵니다.

```
aws iotthingsgraph get-namespace-deletion-status
```

출력:

```
{
   "namespaceArn": "arn:aws:iotthingsgraph:us-west-2:123456789012",
   "namespaceName": "us-west-2/123456789012/default"
   "status": "SUCCEEDED "
}
```

자세한 내용은 AWS IoT 사물 그래프 사용 설명서의 네임스페이스를 참조하세요.

- 자세한 API 내용은 명령 참조GetNamespaceDeletionStatus의 섹션을 참조하세요. AWS CLI

### get-system-instance

다음 코드 예시에서는 get-system-instance을 사용하는 방법을 보여 줍니다.

AWS CLI

시스템 인스턴스를 가져오려면

다음 get-system-instance 예제에서는 시스템 인스턴스에 대한 정의를 가져옵니다.

```
aws iotthingsgraph get-system-instance \
    --id "urn:tdm:us-west-2/123456789012/default:Deployment:Room218"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS IoT 사물 그래프 사용 설명서의 시스템 및 흐름 구성 작업을 참조하세요.

- 자세한 API 내용은 명령 참조GetSystemInstance의 섹션을 참조하세요. AWS CLI

### get-system-template-revisions

다음 코드 예시에서는 get-system-template-revisions을 사용하는 방법을 보여 줍니다.

AWS CLI

시스템에 대한 개정 정보를 가져오려면

다음 get-system-template-revisions 예제에서는 시스템에 대한 개정 정보를 가져옵니다.

```
aws iotthingsgraph get-system-template-revisions \
    --id "urn:tdm:us-west-2/123456789012/default:System:MySystem"
```

출력:

```
{
    "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
        }
```

```
    ]
}
```

자세한 내용은 AWS IoT 사물 그래프 사용 설명서의 시스템 및 흐름 구성 작업을 참조하세요.

- 자세한 API 내용은 명령 참조GetSystemTemplateRevisions의 섹션을 참조하세요. AWS CLI

### get-system-template

다음 코드 예시에서는 get-system-template을 사용하는 방법을 보여 줍니다.

AWS CLI

시스템을 가져오려면

다음 get-system-template 예제에서는 시스템에 대한 정의를 가져옵니다.

```
aws iotthingsgraph get-system-template \
    --id "urn:tdm:us-west-2/123456789012/default:System:MySystem"
```

출력:

```
{
    "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": "{\ntype 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
    }
```

```
}
```

자세한 내용은 AWS IoT 사물 그래프 사용 설명서의 시스템 및 흐름 구성 작업을 참조하세요.

- 자세한 API 내용은 명령 참조GetSystemTemplate의 섹션을 참조하세요. AWS CLI

### `get-upload-status`

다음 코드 예시에서는 `get-upload-status`을 사용하는 방법을 보여 줍니다.

AWS CLI

엔터티 업로드 상태를 가져오려면

다음 `get-upload-status` 예제에서는 엔터티 업로드 작업의 상태를 가져옵니다. 의 값은 `upload-entity-definitions` 작업에서 반환되는 ID 값`MyUploadId`입니다.

```
aws iotthingsgraph get-upload-status \
    --upload-id "MyUploadId"
```

출력:

```
{
    "namespaceName": "us-west-2/123456789012/default",
    "namespaceVersion": 5,
    "uploadId": "f6294f1e-b109-4bbe-9073-f451a2dda2da",
    "uploadStatus": "SUCCEEDED"
}
```

자세한 내용은 AWS IoT 사물 그래프 사용 설명서의 모델링 엔터티를 참조하세요.

- 자세한 API 내용은 명령 참조GetUploadStatus의 섹션을 참조하세요. AWS CLI

### `list-flow-execution-messages`

다음 코드 예시에서는 `list-flow-execution-messages`을 사용하는 방법을 보여 줍니다.

AWS CLI

흐름 실행의 이벤트에 대한 정보를 가져오려면

다음 `list-flow-execution-messages` 예제에서는 흐름 실행의 이벤트에 대한 정보를 가져옵니다.

```
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"
```

출력:

```
{
    "messages": [
        {
         "eventType": "EXECUTION_STARTED",
         "messageId": "f6294f1e-b109-4bbe-9073-f451a2dda2da",
         "payload": "Flow execution started",
         "timestamp": 1559247540.656
        }
    ]
}
```

자세한 내용은 AWS IoT 사물 그래프 사용 설명서의 흐름 작업을 참조하세요.

- 자세한 API 내용은 명령 참조ListFlowExecutionMessages의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스의 모든 태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 AWS IoT Things Graph 리소스의 모든 태그를 나열합니다.

```
aws iotthingsgraph list-tags-for-resource \
    --resource-arn "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/
default/Room218"
```

출력:

```
{
   "tags": [
      {
```

```
        "key": "Type",
        "value": "Residential"
      }
   ]
}
```

자세한 내용은 AWS IoT 사물 그래프 사용 설명서의 IoT 사물 그래프 리소스 태그 지정을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### `search-entities`

다음 코드 예시에서는 `search-entities`을 사용하는 방법을 보여 줍니다.

AWS CLI

엔터티를 검색하려면

다음 `search-entities` 예제에서는 유형의 모든 엔터티를 검색합니다EVENT.

```
aws iotthingsgraph search-entities \
    --entity-types "EVENT"
```

출력:

```
{
    "descriptions": [
        {
            "id": "urn:tdm:aws/examples:Event:MotionSensorEvent",
            "type": "EVENT",
            "definition": {
                "language": "GRAPHQL",
                "text": "##\n# Description of events emitted by motion
 sensor.\n##\ntype 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\ntype
 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"
}
```

자세한 내용은 AWS IoT 사물 그래프 사용 설명서의 IoT 사물 그래프 데이터 모델 참조를 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조SearchEntities의 섹션을 참조하세요. AWS CLI

### `search-flow-executions`

다음 코드 예시에서는 `search-flow-executions`을 사용하는 방법을 보여 줍니다.

AWS CLI

흐름 실행을 검색하려면

다음 `search-flow-executions` 예제에서는 지정된 시스템 인스턴스에서 흐름의 모든 실행을 검색합니다.

```
aws iotthingsgraph search-flow-executions \
    --system-instance-id "urn:tdm:us-west-2/123456789012/default:Deployment:Room218"
```

출력:

```
{
   "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
      }
   ]
}
```

자세한 내용은 AWS IoT 사물 그래프 사용 설명서의 시스템 및 흐름 구성 작업을 참조하세요.

- 자세한 API 내용은 명령 참조SearchFlowExecutions의 섹션을 참조하세요. AWS CLI

### `search-flow-templates`

다음 코드 예시에서는 `search-flow-templates`을 사용하는 방법을 보여 줍니다.

AWS CLI

흐름(또는 워크플로)을 검색하려면

다음 `search-flow-templates` 예제에서는 카메라 디바이스 모델을 포함하는 모든 흐름(워크플로)을 검색합니다.

```
aws iotthingsgraph search-flow-templates \
    --filters name="DEVICE_MODEL_ID",value="urn:tdm:aws/examples:DeviceModel:Camera"
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 AWS IoT 사물 그래프 사용 설명서의 흐름 작업을 참조하세요.

- 자세한 API 내용은 명령 참조SearchFlowTemplates의 섹션을 참조하세요. AWS CLI

### `search-system-instances`

다음 코드 예시에서는 `search-system-instances`을 사용하는 방법을 보여 줍니다.

AWS CLI

시스템 인스턴스를 검색하려면

다음 `search-system-instances` 예제에서는 지정된 시스템을 포함하는 모든 시스템 인스턴스를 검색합니다.

```
aws iotthingsgraph search-system-instances \
    --filters name="SYSTEM_TEMPLATE_ID",value="urn:tdm:us-west-2/123456789012/
default:System:SecurityFlow"
```

출력:

```
{
    "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
        }
```

```
    ]
}
```

자세한 내용은 AWS IoT 사물 그래프 사용 설명서의 시스템 및 흐름 구성 작업을 참조하세요.

- 자세한 API 내용은 명령 참조SearchSystemInstances의 섹션을 참조하세요. AWS CLI

### `search-system-templates`

다음 코드 예시에서는 `search-system-templates`을 사용하는 방법을 보여 줍니다.

AWS CLI

시스템을 검색하려면

다음 `search-system-templates` 예제에서는 지정된 흐름이 포함된 모든 시스템을 검색합니다.

```
aws iotthingsgraph search-system-templates \
    --filters name="FLOW_TEMPLATE_ID",value="urn:tdm:us-west-2/123456789012/
default:Workflow:SecurityFlow"
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 AWS IoT 사물 그래프 사용 설명서의 흐름 작업을 참조하세요.

- 자세한 API 내용은 명령 참조SearchSystemTemplates의 섹션을 참조하세요. AWS CLI

### `search-things`

다음 코드 예시에서는 `search-things`을 사용하는 방법을 보여 줍니다.

AWS CLI

디바이스 및 디바이스 모델과 연결된 사물을 검색하려면

다음 `search-things` 예제에서는 HCSR501MotionSensor device와 연결된 모든 항목을 검색합니다.

```
aws iotthingsgraph search-things \
    --entity-id "urn:tdm:aws/examples:Device:HCSR501MotionSensor"
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS IoT 사물 그래프 사용 설명서의 모델 생성 및 업로드를 참조하세요.

- 자세한 API 내용은 명령 참조SearchThings의 섹션을 참조하세요. AWS CLI

### `tag-resource`

다음 코드 예시에서는 `tag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 대한 태그를 생성하려면

다음 `tag-resource` 예제에서는 지정된 리소스에 대한 태그를 생성합니다.

```
aws iotthingsgraph tag-resource \
    --resource-arn "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/
default/Room218" \
```

```
    --tags key="Type",value="Residential"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 사물 그래프 사용 설명서의 IoT 사물 그래프 리소스 태그 지정을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### undeploy-system-instance

다음 코드 예시에서는 undeploy-system-instance을 사용하는 방법을 보여 줍니다.

AWS CLI

대상에서 시스템 인스턴스를 배포 취소하려면

다음 undeploy-system-instance 예제에서는 대상에서 시스템 인스턴스를 제거합니다.

```
aws iotthingsgraph undeploy-system-instance \
    --id "urn:tdm:us-west-2/123456789012/default:Deployment:Room215"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS IoT 사물 그래프 사용 설명서의 IoT 사물 그래프 개체, 흐름, 시스템 및 배포의 수명 주기 관리를 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조UndeploySystemInstance의 섹션을 참조하세요. AWS CLI

### untag-resource

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스의 태그를 제거하려면

다음 untag-resource 예제에서는 지정된 리소스에 대한 태그를 제거합니다.

```
aws iotthingsgraph untag-resource \
    --resource-arn "arn:aws:iotthingsgraph:us-west-2:123456789012:Deployment/
default/Room218" \
    --tag-keys "Type"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 사물 그래프 사용 설명서의 IoT 사물 그래프 리소스 태그 지정을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-flow-template

다음 코드 예시에서는 update-flow-template을 사용하는 방법을 보여 줍니다.

AWS CLI

흐름을 업데이트하려면

다음 update-flow-template 예제에서는 흐름(워크플로)을 업데이트합니다. 의 값은 흐름을 모델링하는 GraphQLMyFlowDefinition입니다.

```
aws iotthingsgraph update-flow-template \
    --id "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow" \
    --definition language=GRAPHQL,text="MyFlowDefinition"
```

출력:

```
{
    "summary": {
        "createdAt": 1559248067.545,
        "id": "urn:tdm:us-west-2/123456789012/default:Workflow:MyFlow",
        "revisionNumber": 2
    }
}
```

자세한 내용은 AWS IoT 사물 그래프 사용 설명서의 흐름 작업을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateFlowTemplate의 섹션을 참조하세요. AWS CLI

### update-system-template

다음 코드 예시에서는 update-system-template을 사용하는 방법을 보여 줍니다.

AWS CLI

시스템을 업데이트하려면

다음 update-system-template 예제에서는 시스템을 업데이트합니다. 의 값은 시스템을 모델링하는 GraphQLMySystemDefinition입니다.

```
aws iotthingsgraph update-system-template \
    --id "urn:tdm:us-west-2/123456789012/default:System:MySystem" \
    --definition language=GRAPHQL,text="MySystemDefinition"
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 AWS IoT 사물 그래프 사용 설명서의 시스템 생성을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateSystemTemplate의 섹션을 참조하세요. AWS CLI

### upload-entity-definitions

다음 코드 예시에서는 upload-entity-definitions을 사용하는 방법을 보여 줍니다.

AWS CLI

개체 정의를 업로드하려면

다음 upload-entity-definitions 예제에서는 네임스페이스에 엔터티 정의를 업로드합니다. 의 값은 엔터티를 모델링하는 GraphQLMyEntityDefinitions입니다.

```
aws iotthingsgraph upload-entity-definitions \
    --document language=GRAPHQL,text="MyEntityDefinitions"
```

출력:

```
{
    "uploadId": "f6294f1e-b109-4bbe-9073-f451a2dda2da"
}
```

자세한 내용은 AWS IoT 사물 그래프 사용 설명서의 모델링 엔터티를 참조하세요.

- 자세한 API 내용은 명령 참조UploadEntityDefinitions의 섹션을 참조하세요. AWS CLI

## AWS IoT 무선 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS IoT 무선.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

작업

### `associate-aws-account-with-partner-account`

다음 코드 예시에서는 `associate-aws-account-with-partner-account`을 사용하는 방법을 보여 줍니다.

AWS CLI

파트너 계정을 AWS 계정과 연결하려면

다음 `associate-aws-account-with-partner-account` 예제는 다음 Sidewalk 계정 자격 증명을 AWS 계정과 연결합니다.

```
aws iotwireless associate-aws-account-with-partner-account \
    --sidewalk
 AmazonId="12345678901234",AppServerPrivateKey="a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78
```

출력:

```
{
    "Sidewalk": {
        "AmazonId": "12345678901234",
        "AppServerPrivateKey":
 "a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78a1b234c56d7e890a1234"
    }
}
```

자세한 내용은 AWS IoT 개발자 안내서의 IoT Core용 Amazon Sidewalk 통합AWS 을 참조하세요 IoT.

- 자세한 API 내용은 명령 참조AssociateAwsAccountWithPartnerAccount의 섹션을 참조하세요. AWS CLI

### `associate-wireless-device-with-thing`

다음 코드 예시에서는 `associate-wireless-device-with-thing`을 사용하는 방법을 보여 줍니다.

AWS CLI

무선 디바이스에 사물을 연결하려면

다음 associate-wireless-device-with-thing 예제는 지정된 ID가 있는 무선 디바이스에 사물을 연결합니다.

```
aws iotwireless associate-wireless-device-with-thing \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \
    --thing-arn "arn:aws:iot:us-east-1:123456789012:thing/MyIoTWirelessThing"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 게이트웨이 및 무선 디바이스 추가 LoRaWANAWS 를 참조하세요 IoT.

- 자세한 API 내용은 명령 참조AssociateWirelessDeviceWithThing의 섹션을 참조하세요. AWS CLI

### associate-wireless-gateway-with-certificate

다음 코드 예시에서는 associate-wireless-gateway-with-certificate을 사용하는 방법을 보여 줍니다.

AWS CLI

인증서를 무선 게이트웨이에 연결하려면

다음은 무선 게이트웨이를 인증서와 associate-wireless-gateway-with-certificate 연결합니다.

```
aws iotwireless associate-wireless-gateway-with-certificate \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \
    --iot-certificate-
id "a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78a1b234c56d7e890a1234"
```

출력:

```
{
    "IotCertificateId":
 "a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78a1b234c56d7e890a1234"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 게이트웨이 및 무선 디바이스 추가 LoRaWANAWS 를 참조하세요 IoT.

- 자세한 API 내용은 명령 참조AssociateWirelessGatewayWithCertificate의 섹션을 참조하세요. AWS CLI

### associate-wireless-gateway-with-thing

다음 코드 예시에서는 associate-wireless-gateway-with-thing을 사용하는 방법을 보여 줍니다.

AWS CLI

사물을 무선 게이트웨이에 연결하려면

다음 associate-wireless-gateway-with-thing 예제에서는 사물을 무선 게이트웨이에 연결합니다.

```
aws iotwireless associate-wireless-gateway-with-thing \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \
    --thing-arn "arn:aws:iot:us-east-1:123456789012:thing/MyIoTWirelessThing"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 게이트웨이 및 무선 디바이스 추가 LoRaWANAWS 를 참조하세요 IoT.

- 자세한 API 내용은 명령 참조AssociateWirelessGatewayWithThing의 섹션을 참조하세요. AWS CLI

### create-destination

다음 코드 예시에서는 create-destination을 사용하는 방법을 보여 줍니다.

AWS CLI

IoT 무선 대상을 생성하려면

다음 create-destination 예제에서는 디바이스 메시지를 AWS IoT 규칙에 매핑하기 위한 대상을 생성합니다. 이 명령을 실행하기 전에 AWS IoT 규칙으로 데이터를 전송하는 데 필요한 권한을 LoRaWAN AWS IoT Core에 부여하는 IAM 역할을 생성했어야 합니다.

```
aws iotwireless create-destination \
    --name IoTWirelessDestination \
```

```
    --expression-type RuleName \
    --expression IoTWirelessRule \
    --role-arn arn:aws:iam::123456789012:role/IoTWirelessDestinationRole
```

출력:

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:Destination/
IoTWirelessDestination",
    "Name": "IoTWirelessDestination"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 대상 추가 LoRaWANAWS 를 참조하세요 IoT.

- 자세한 API 내용은 명령 참조CreateDestination의 섹션을 참조하세요. AWS CLI

### `create-device-profile`

다음 코드 예시에서는 `create-device-profile`을 사용하는 방법을 보여 줍니다.

AWS CLI

새 디바이스 프로필을 생성하려면

다음 `create-device-profile` 예제에서는 새 IoT 무선 디바이스 프로파일을 생성합니다.

```
aws iotwireless create-device-profile
```

출력:

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:DeviceProfile/12345678-
a1b2-3c45-67d8-e90fa1b2c34d",
    "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 프로필 추가 LoRaWANAWS 를 참조하세요 IoT.

- 자세한 API 내용은 명령 참조CreateDeviceProfile의 섹션을 참조하세요. AWS CLI

### create-service-profile

다음 코드 예시에서는 create-service-profile을 사용하는 방법을 보여 줍니다.

AWS CLI

새 서비스 프로필을 생성하려면

다음 create-service-profile 예제에서는 새 IoT 무선 서비스 프로파일을 생성합니다.

```
aws iotwireless create-service-profile
```

출력:

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:ServiceProfile/12345678-
a1b2-3c45-67d8-e90fa1b2c34d",
    "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 프로필 추가 LoRaWANAWS 를 참조하세요 IoT.

- 자세한 API 내용은 명령 참조CreateServiceProfile의 섹션을 참조하세요. AWS CLI

### create-wireless-device

다음 코드 예시에서는 create-wireless-device을 사용하는 방법을 보여 줍니다.

AWS CLI

IoT 무선 디바이스를 생성하려면

다음 create-wireless-device 예제에서는 유형의 무선 디바이스 리소스를 생성합니다 LoRaWAN.

```
aws iotwireless create-wireless-device \
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
    "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
}
```

출력:

```
{
    "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:WirelessDevice/1ffd32c8-8130-4194-96df-622f072a315f",
    "Id": "1ffd32c8-8130-4194-96df-622f072a315f"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 디바이스 및 게이트웨이 연결을 LoRaWAN AWS 참조하세요 IoT.

- 자세한 API 내용은 명령 참조CreateWirelessDevice의 섹션을 참조하세요. AWS CLI

### `create-wireless-gateway-task-definition`

다음 코드 예시에서는 `create-wireless-gateway-task-definition`을 사용하는 방법을 보여 줍니다.

AWS CLI

무선 게이트웨이 작업 정의를 생성하려면

다음은 지정된 현재 버전의 모든 게이트웨이에 대해 이 작업 정의를 사용하여 작업을 `create-wireless-gateway-task-definition` 자동으로 생성합니다.

```
aws iotwireless create-wireless-gateway-task-definition \
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
    "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"
            }
        }
    }
}
```

출력:

```
{
    "Id": "b7d3baad-25c7-35e7-a4e1-1683a0d61da9"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 디바이스 및 게이트웨이 연결을 LoRaWAN AWS 참조하세요 IoT.

- 자세한 API 내용은 명령 참조CreateWirelessGatewayTaskDefinition의 섹션을 참조하세요. AWS CLI

### create-wireless-gateway-task

다음 코드 예시에서는 create-wireless-gateway-task을 사용하는 방법을 보여 줍니다.

AWS CLI

무선 게이트웨이에 대한 작업을 생성하려면

다음 `create-wireless-gateway-task` 예제에서는 무선 게이트웨이에 대한 작업을 생성합니다.

```
aws iotwireless create-wireless-gateway-task \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \
    --wireless-gateway-task-definition-id "aa000102-0304-b0cd-ef56-a1b23cde456a"
```

출력:

```
{
    "WirelessGatewayTaskDefinitionId": "aa204003-0604-30fb-ac82-a4f95aaf450a",
    "Status": "Success"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 디바이스 및 게이트웨이 연결을 LoRaWAN AWS 참조하세요 IoT.

- 자세한 API 내용은 명령 참조CreateWirelessGatewayTask의 섹션을 참조하세요. AWS CLI

### `create-wireless-gateway`

다음 코드 예시에서는 `create-wireless-gateway`을 사용하는 방법을 보여 줍니다.

AWS CLI

무선 게이트웨이를 생성하려면

다음 `create-wireless-gateway` 예제에서는 무선 LoRaWAN 디바이스 게이트웨이를 생성합니다.

```
aws iotwireless create-wireless-gateway \
    --lorawan GatewayEui="a1b2c3d4567890ab",RfRegion="US915" \
    --name "myFirstLoRaWANGateway" \
    --description "Using my first LoRaWAN gateway"
```

출력:

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:WirelessGateway/12345678-
a1b2-3c45-67d8-e90fa1b2c34d",
    "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 디바이스 및 게이트웨이 연결을 LoRaWAN AWS 참조하세요 IoT.

- 자세한 API 내용은 명령 참조CreateWirelessGateway의 섹션을 참조하세요. AWS CLI

### delete-destination

다음 코드 예시에서는 delete-destination을 사용하는 방법을 보여 줍니다.

AWS CLI

IoT 무선 대상을 삭제하려면

다음 delete-destination 예제에서는 IoTWirelessDestination 생성한 이름을 사용하여 무선 대상 리소스를 삭제합니다.

```
aws iotwireless delete-destination \
    --name "IoTWirelessDestination"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 대상 추가 LoRaWANAWS 를 참조하세요 IoT.

- 자세한 API 내용은 명령 참조DeleteDestination의 섹션을 참조하세요. AWS CLI

### delete-device-profile

다음 코드 예시에서는 delete-device-profile을 사용하는 방법을 보여 줍니다.

AWS CLI

디바이스 프로필을 삭제하려면

다음 delete-device-profile 예제에서는 생성한 지정된 ID가 있는 디바이스 프로파일을 삭제합니다.

```
aws iotwireless delete-device-profile \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 프로필 추가 LoRaWANAWS 를 참조하세요 IoT.

- 자세한 API 내용은 명령 참조DeleteDeviceProfile의 섹션을 참조하세요. AWS CLI

### delete-service-profile

다음 코드 예시에서는 delete-service-profile을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스 프로필을 삭제하려면

다음 delete-service-profile 예제에서는 생성한 지정된 ID가 있는 서비스 프로파일을 삭제합니다.

```
aws iotwireless delete-service-profile \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 프로필 추가 LoRaWANAWS 를 참조하세요 IoT.

- 자세한 API 내용은 명령 참조DeleteServiceProfile의 섹션을 참조하세요. AWS CLI

### delete-wireless-device

다음 코드 예시에서는 delete-wireless-device을 사용하는 방법을 보여 줍니다.

AWS CLI

무선 디바이스를 삭제하려면

다음 delete-wireless-device 예제에서는 지정된 ID가 있는 무선 디바이스를 삭제합니다.

```
aws iotwireless delete-wireless-device \
```

```
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 디바이스 및 게이트웨이 연결을 LoRaWAN AWS 참조하세요 IoT.

- 자세한 API 내용은 명령 참조DeleteWirelessDevice의 섹션을 참조하세요. AWS CLI

### `delete-wireless-gateway-task-definition`

다음 코드 예시에서는 `delete-wireless-gateway-task-definition`을 사용하는 방법을 보여 줍니다.

AWS CLI

무선 게이트웨이 작업 정의를 삭제하려면

다음 `delete-wireless-gateway-task-definition` 예제에서는 다음 ID로 생성한 무선 게이트웨이 작업 정의를 삭제합니다.

```
aws iotwireless delete-wireless-gateway-task-definition \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 디바이스 및 게이트웨이 연결을 LoRaWAN AWS 참조하세요 IoT.

- 자세한 API 내용은 명령 참조DeleteWirelessGatewayTaskDefinition의 섹션을 참조하세요. AWS CLI

### `delete-wireless-gateway-task`

다음 코드 예시에서는 `delete-wireless-gateway-task`을 사용하는 방법을 보여 줍니다.

AWS CLI

무선 게이트웨이 작업을 삭제하려면

다음 `delete-wireless-gateway-task` 예제에서는 지정된 ID가 있는 무선 게이트웨이 작업을 삭제합니다.

```
aws iotwireless delete-wireless-gateway-task \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 디바이스 및 게이트웨이 연결을 LoRaWAN AWS 참조하세요 IoT.

- 자세한 API 내용은 명령 참조DeleteWirelessGatewayTask의 섹션을 참조하세요. AWS CLI

### `delete-wireless-gateway`

다음 코드 예시에서는 `delete-wireless-gateway`을 사용하는 방법을 보여 줍니다.

AWS CLI

무선 게이트웨이를 삭제하려면

다음 `delete-wireless-gateway` 예제에서는 지정된 ID가 있는 무선 게이트웨이를 삭제합니다.

```
aws iotwireless delete-wireless-gateway \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 디바이스 및 게이트웨이 연결을 LoRaWAN AWS 참조하세요 IoT.

- 자세한 API 내용은 명령 참조DeleteWirelessGateway의 섹션을 참조하세요. AWS CLI

### `disassociate-aws-account-from-partner-account`

다음 코드 예시에서는 `disassociate-aws-account-from-partner-account`을 사용하는 방법을 보여 줍니다.

AWS CLI

계정에서 파트너 계정 연결을 해제하려면 AWS

다음 `disassociate-aws-account-from-partner-account` 예제에서는 현재 연결된 계정에서 파트너 AWS 계정의 연결을 해제합니다.

```
aws iotwireless disassociate-aws-account-from-partner-account \
    --partner-account-id "12345678901234" \
    --partner-type "Sidewalk"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 게이트웨이 및 무선 디바이스 추가 LoRaWANAWS 를 참조하세요 IoT.

- 자세한 API 내용은 명령 참조DisassociateAwsAccountFromPartnerAccount의 섹션을 참조하세요. AWS CLI

### disassociate-wireless-device-from-thing

다음 코드 예시에서는 disassociate-wireless-device-from-thing을 사용하는 방법을 보여 줍니다.

AWS CLI

무선 디바이스에서 사물의 연결을 해제하려면

다음 disassociate-wireless-device-from-thing 예제에서는 무선 디바이스를 현재 연결된 사물과 연결 해제합니다.

```
aws iotwireless disassociate-wireless-device-from-thing \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 게이트웨이 및 무선 디바이스 추가 LoRaWANAWS 를 참조하세요 IoT.

- 자세한 API 내용은 명령 참조DisassociateWirelessDeviceFromThing의 섹션을 참조하세요. AWS CLI

### disassociate-wireless-gateway-from-certificate

다음 코드 예시에서는 disassociate-wireless-gateway-from-certificate을 사용하는 방법을 보여 줍니다.

AWS CLI

무선 게이트웨이에서 인증서 연결을 해제하려면

다음은 현재 연결된 인증서에서 무선 게이트웨이의 연결을 `disassociate-wireless-gateway-from-certificate` 해제합니다.

```
aws iotwireless disassociate-wireless-gateway-from-certificate \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 게이트웨이 및 무선 디바이스 추가 LoRaWANAWS 를 참조하세요 IoT.

- 자세한 API 내용은 명령 참조DisassociateWirelessGatewayFromCertificate의 섹션을 참조하세요. AWS CLI

### `disassociate-wireless-gateway-from-thing`

다음 코드 예시에서는 `disassociate-wireless-gateway-from-thing`을 사용하는 방법을 보여 줍니다.

AWS CLI

무선 게이트웨이에서 사물의 연결을 해제하려면

다음 `disassociate-wireless-gateway-from-thing` 예제에서는 무선 게이트웨이를 현재 연결된 사물과 연결 해제합니다.

```
aws iotwireless disassociate-wireless-gateway-from-thing \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 게이트웨이 및 무선 디바이스 추가 LoRaWANAWS 를 참조하세요 IoT.

- 자세한 API 내용은 명령 참조DisassociateWirelessGatewayFromThing의 섹션을 참조하세요. AWS CLI

### get-destination

다음 코드 예시에서는 get-destination을 사용하는 방법을 보여 줍니다.

AWS CLI

IoT 무선 대상에 대한 정보를 가져오려면

다음 get-destination 예제에서는 IoTWirelessDestination 생성한 이름을 사용하여 대상 리소스에 대한 정보를 가져옵니다.

```
aws iotwireless get-destination \
    --name "IoTWirelessDestination"
```

출력:

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:Destination/
IoTWirelessDestination",
    "Name": "IoTWirelessDestination",
    "Expression": "IoTWirelessRule",
    "ExpressionType": "RuleName",
    "RoleArn": "arn:aws:iam::123456789012:role/IoTWirelessDestinationRole"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 대상 추가 LoRaWANAWS 를 참조하세요 IoT.

- 자세한 API 내용은 명령 참조GetDestination의 섹션을 참조하세요. AWS CLI

### get-device-profile

다음 코드 예시에서는 get-device-profile을 사용하는 방법을 보여 줍니다.

AWS CLI

디바이스 프로파일에 대한 정보를 가져오려면

다음 get-device-profile 예제에서는 생성한 지정된 ID를 사용하여 디바이스 프로파일에 대한 정보를 가져옵니다.

```
aws iotwireless get-device-profile \
```

```
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 프로필 추가 LoRaWANAWS 를 참조하세요 IoT.

- 자세한 API 내용은 명령 참조GetDeviceProfile의 섹션을 참조하세요. AWS CLI

### get-partner-account

다음 코드 예시에서는 get-partner-account을 사용하는 방법을 보여 줍니다.

AWS CLI

파트너 계정 정보를 가져오려면

다음 get-partner-account 예제에서는 다음 ID를 가진 Sidewalk 계정에 대한 정보를 가져옵니다.

```
aws iotwireless get-partner-account \
    --partner-account-id "12345678901234" \
    --partner-type "Sidewalk"
```

출력:

```
{
    "Sidewalk": {
        "AmazonId": "12345678901234",
        "Fingerprint":
 "a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78a1b234c56d7e890a1234"
    },
    "AccountLinked": false
}
```

자세한 내용은 AWS IoT 개발자 안내서의 IoT Core용 Amazon Sidewalk 통합AWS 을 참조하세요 IoT.

- 자세한 API 내용은 명령 참조GetPartnerAccount의 섹션을 참조하세요. AWS CLI

### get-service-endpoint

다음 코드 예시에서는 get-service-endpoint을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스 엔드포인트를 가져오려면

다음 get-service-endpoint 예제에서는 CUPS 프로토콜의 계정별 엔드포인트를 가져옵니다.

```
aws iotwireless get-service-endpoint
```

출력:

```
{
    "ServiceType": "CUPS",
    "ServiceEndpoint": "https://A1RMKZ37ACAGOT.cups.lorawan.us-
east-1.amazonaws.com:443",
    "ServerTrust": "-----BEGIN CERTIFICATE-----\n
    MIIESTCCAzGgAwIBAgITBn+UV4WH6Kx33rJTMlu8mYtWDTANBgkqhkiG9w0BAQsF\n
    ADA5MQswCQYDVQQGEwJVUzEPMA0GA1UEChMGQW1hem9uMRkwFwYDVQQDExBBbWF6\n
    b24gUm9vdCBDQSAxMB4XDTE1MTAyMjAwMDAwMFoXDTI1MTAxOTAwMDAwMFowRjEL\n
    MAkGA1UEBhMCVVMxDzANBgNVBAoTBkFtYXpvbjEVMBMGA1UECxMMU2VydmVyIENB\n
    IDFCMQ8wDQYDVQQDEwZBbWF6b24wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEK\n
    AoIBAQDCThZn3c68asg3Wuw6MLAd5tES6BIoSMzoKcG5blPVo+sDORrMd4f2AbnZ\n
    cMzPa43j4wNxhplty6aUKk4T1qe9BOwKFjwK6zmxxLVYo7bHViXsPlJ6qOMpFge5\n
    blDP+18x+B26A0piiQOuPkfyDyeR4xQghfj66Yo19V+emU3nazfvpFA+ROz6WoVm\n
    B5x+F2pV8xeKNR7u6azDdU5YVX1TawprmxRC1+WsAYmz6qP+z8ArDITC2FMVy2fw\n
```

```
    0IjKOtEXc/VfmtTFch5+AfGYMGMqqvJ6LcXiAhqG5TI+Dr0RtM88k+8XUBCeQ8IG\n
    KuANaL7TiItKZYxK1MMuTJtV9IblAgMBAAGjggE7MIIBNzASBgNVHRMBAf8ECDAG\n
    AQH/AgEAMA4GA1UdDwEB/wQEAwIBhjAdBgNVHQ4EFgQUWaRmBlKge5WSPKOUByeW\n
    dFv5PdAwHwYDVR0jBBgwFoAUhBjMhTTsvAyUlC4IWZzHshBOCggwewYIKwYBBQUH\n
    AQEEbzBtMC8GCCsGAQUFBzABhiNodHRwOi8vb2NzcC5yb290Y2ExLmFtYXpvbnRy\n
    dXN0LmNvbTA6BggrBgEFBQcwAoYuaHR0cDovL2NydC5yb290Y2ExLmFtYXpvbnRy\n
    dXN0LmNvbS9yb290Y2ExLmNlcjA/BgNVHR8EODA2MDSgMqAwhi5odHRwOi8vY3Js\n
    LnJvb3RjYTEuYW1hem9udHJ1c3QuY29tL3Jvb3RjYTEuY3JsMBMGA1UdIAQMMAow\n
    CAYGZ4EMAQIBMA0GCSqGSIb3DQEBCwUAA4IBAQCFkr41u3nPo4FCHOTjY3NTOVI1\n
    59Gt/a6ZiqyJEi+752+a1U5y6iAwYfmXss2lJwJFqMp2PphKg5625kXg8kP2CN5t\n
    6G7bMQcT8C8xDZNtYTd7WPD8UZiRKAJPBXa30/AbwuZe0GaFEQ8ugcYQgSn+IGBI\n
    8/LwhBNTZTUVEWuCUUBVV18YtbAiPq3yXqMB48Oz+ctBWuZSkbvkNodPLamkB2g1\n
    upRyzQ7qDn1X8nn8N8V7YJ6y68AtkHcNSRAnpTitxBKjtKPISLMVCx7i4hncxHZS\n
    yLyKQXhw2W2Xs0qLeC1etA+jTGDK4UfLeC0SF7FSi8o5LL21L8IzApar2pR/\n
    -----END CERTIFICATE-----\n"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 디바이스 및 게이트웨이 연결을 LoRaWAN AWS 참조하세요 IoT.

- 자세한 API 내용은 명령 참조GetServiceEndpoint의 섹션을 참조하세요. AWS CLI

### get-service-profile

다음 코드 예시에서는 get-service-profile을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스 프로파일에 대한 정보를 가져오려면

다음 get-service-profile 예제에서는 생성한 지정된 ID를 사용하여 서비스 프로파일에 대한 정보를 가져옵니다.

```
aws iotwireless get-service-profile \
    --id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 프로필 추가 LoRaWANAWS 를 참조하세요 IoT.

- 자세한 API 내용은 명령 참조GetServiceProfile의 섹션을 참조하세요. AWS CLI

### get-wireless-device-statistics

다음 코드 예시에서는 get-wireless-device-statistics을 사용하는 방법을 보여 줍니다.

AWS CLI

무선 디바이스에 대한 작동 정보를 가져오려면

다음 get-wireless-device-statistics 예제에서는 무선 디바이스에 대한 작동 정보를 가져옵니다.

```
aws iotwireless get-wireless-device-statistics \
    --wireless-device-id "1ffd32c8-8130-4194-96df-622f072a315f"
```

출력:

```
{
```

```
    "WirelessDeviceId": "1ffd32c8-8130-4194-96df-622f072a315f"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 디바이스 및 게이트웨이 연결을 LoRaWAN AWS 참조하세요 IoT.

- 자세한 API 내용은 명령 참조GetWirelessDeviceStatistics의 섹션을 참조하세요. AWS CLI

### get-wireless-device

다음 코드 예시에서는 get-wireless-device을 사용하는 방법을 보여 줍니다.

AWS CLI

무선 디바이스에 대한 정보를 가져오려면

다음 get-wireless-device 예제에서는 AWS 계정에서 사용 가능한 위젯을 나열합니다.

```
aws iotwireless get-wireless-device \
    --identifier "1ffd32c8-8130-4194-96df-622f072a315f" \
    --identifier-type WirelessDeviceID
```

출력:

```
{
    "Name": "myLoRaWANDevice",
    "ThingArn": "arn:aws:iot:us-east-1:123456789012:thing/44b87eb4-9bce-423d-
b5fc-973f5ecc358b",
    "DestinationName": "IoTWirelessDestination",
    "Id": "1ffd32c8-8130-4194-96df-622f072a315f",
    "ThingName": "44b87eb4-9bce-423d-b5fc-973f5ecc358b",
    "Type": "LoRaWAN",
    "LoRaWAN": {
        "DeviceProfileId": "ab0c23d3-b001-45ef-6a01-2bc3de4f5333",
        "ServiceProfileId": "fe98dc76-cd12-001e-2d34-5550432da100",
        "OtaaV1_1": {
            "AppKey": "3f4ca100e2fc675ea123f4eb12c4a012",
            "JoinEui": "b4c231a359bc2e3d",
            "NwkKey": "01c3f004a2d6efffe32c4eda14bcd2b4"
        },
        "DevEui": "ac12efc654d23fc2"
    },
```

```
    "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:WirelessDevice/1ffd32c8-8130-4194-96df-622f072a315f",
    "Description": "My LoRaWAN wireless device"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 디바이스 및 게이트웨이 연결을 LoRaWAN AWS 참조하세요 IoT.

- 자세한 API 내용은 명령 참조GetWirelessDevice의 섹션을 참조하세요. AWS CLI

### get-wireless-gateway-certificate

다음 코드 예시에서는 get-wireless-gateway-certificate을 사용하는 방법을 보여 줍니다.

AWS CLI

무선 게이트웨이와 연결된 인증서의 ID를 가져오려면

다음 get-wireless-gateway-certificate 예제에서는 지정된 ID가 있는 무선 게이트웨이와 연결된 인증서 ID를 가져옵니다.

```
aws iotwireless get-wireless-gateway-certificate \
    --id "6c44ab31-8b4d-407a-bed3-19b6c7cda551"
```

출력:

```
{
    "IotCertificateId":
 "8ea4aeae3db34c78cce75d9abd830356869ead6972997e0603e5fd032c804b6f"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 디바이스 및 게이트웨이 연결을 LoRaWAN AWS 참조하세요 IoT.

- 자세한 API 내용은 명령 참조GetWirelessGatewayCertificate의 섹션을 참조하세요. AWS CLI

### get-wireless-gateway-firmware-information

다음 코드 예시에서는 get-wireless-gateway-firmware-information을 사용하는 방법을 보여 줍니다.

AWS CLI

무선 게이트웨이에 대한 펌웨어 정보를 가져오려면

다음 `get-wireless-gateway-firmware-information` 예제에서는 무선 게이트웨이에 대한 펌웨어 버전 및 기타 정보를 가져옵니다.

```
aws iotwireless get-wireless-gateway-firmware-information \
    --id "3039b406-5cc9-4307-925b-9948c63da25b"
```

출력:

```
{
    "LoRaWAN" :{
        "CurrentVersion" :{
            "PackageVersion" : "1.0.0",
            "Station" : "2.0.5",
            "Model" : "linux"
        }
    }
}
```

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 디바이스 및 게이트웨이 연결을 LoRaWAN AWS 참조하세요 IoT.

- 자세한 API 내용은 명령 참조GetWirelessGatewayFirmwareInformation의 섹션을 참조하세요. AWS CLI

### `get-wireless-gateway-statistics`

다음 코드 예시에서는 `get-wireless-gateway-statistics`을 사용하는 방법을 보여 줍니다.

AWS CLI

무선 게이트웨이에 대한 작동 정보를 가져오려면

다음 `get-wireless-gateway-statistics` 예제에서는 무선 게이트웨이에 대한 작동 정보를 가져옵니다.

```
aws iotwireless get-wireless-gateway-statistics \
```

```
    --wireless-gateway-id "3039b406-5cc9-4307-925b-9948c63da25b"
```

출력:

```
{
    "WirelessGatewayId": "3039b406-5cc9-4307-925b-9948c63da25b"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 디바이스 및 게이트웨이 연결을 LoRaWAN AWS 참조하세요 IoT.

- 자세한 API 내용은 명령 참조GetWirelessGatewayStatistics의 섹션을 참조하세요. AWS CLI

### get-wireless-gateway-task-definition

다음 코드 예시에서는 get-wireless-gateway-task-definition을 사용하는 방법을 보여 줍니다.

AWS CLI

무선 게이트웨이 작업 정의에 대한 정보를 가져오려면

다음 get-wireless-gateway-task-definition 예제에서는 지정된 ID를 사용하여 무선 태스크 정의에 대한 정보를 가져옵니다.

```
aws iotwireless get-wireless-gateway-task-definition \
    --id "b7d3baad-25c7-35e7-a4e1-1683a0d61da9"
```

출력:

```
{
    "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"
            }
        }
    }
}
```

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 디바이스 및 게이트웨이 연결을 LoRaWAN AWS 참조하세요 IoT.

- 자세한 API 내용은 명령 참조GetWirelessGatewayTaskDefinition의 섹션을 참조하세요. AWS CLI

### get-wireless-gateway-task

다음 코드 예시에서는 get-wireless-gateway-task을 사용하는 방법을 보여 줍니다.

AWS CLI

무선 게이트웨이 작업에 대한 정보를 가져오려면

다음 get-wireless-gateway-task 예제에서는 지정된 ID를 사용하여 무선 게이트웨이 작업에 대한 정보를 가져옵니다.

```
aws iotwireless get-wireless-gateway-task \
    --id "11693a46-6866-47c3-a031-c9a616e7644b"
```

출력:

```
{
    "WirelessGatewayId": "6c44ab31-8b4d-407a-bed3-19b6c7cda551",
    "WirelessGatewayTaskDefinitionId": "b7d3baad-25c7-35e7-a4e1-1683a0d61da9",
    "Status": "Success"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 디바이스 및 게이트웨이 연결을 LoRaWAN AWS 참조하세요 IoT.

- 자세한 API 내용은 명령 참조GetWirelessGatewayTask의 섹션을 참조하세요. AWS CLI

### get-wireless-gateway

다음 코드 예시에서는 get-wireless-gateway을 사용하는 방법을 보여 줍니다.

AWS CLI

무선 게이트웨이에 대한 정보를 가져오려면

다음 get-wireless-gateway 예제에서는 무선 게이트웨이 에 대한 정보를 가져옵니다myFirstLoRaWANGateway.

```
aws iotwireless get-wireless-gateway \
    --identifier "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \
    --identifier-type WirelessGatewayId
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 디바이스 및 게이트웨이 연결을 LoRaWAN AWS 참조하세요 IoT.

- 자세한 API 내용은 명령 참조GetWirelessGateway의 섹션을 참조하세요. AWS CLI

### list-destinations

다음 코드 예시에서는 list-destinations을 사용하는 방법을 보여 줍니다.

AWS CLI

무선 대상을 나열하려면

다음 `list-destinations` 예제에서는 AWS 계정에 등록된 사용 가능한 대상을 나열합니다.

```
aws iotwireless list-destinations
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 대상 추가 LoRaWANAWS 를 참조하세요 IoT.

- 자세한 API 내용은 명령 참조ListDestinations의 섹션을 참조하세요. AWS CLI

### `list-device-profiles`

다음 코드 예시에서는 `list-device-profiles`을 사용하는 방법을 보여 줍니다.

AWS CLI

디바이스 프로필을 나열하려면

다음 list-device-profiles 예제에서는 AWS 계정에 등록된 사용 가능한 디바이스 프로파일을 나열합니다.

```
aws iotwireless list-device-profiles
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 프로필 추가 LoRaWANAWS 를 참조하세요 IoT.

- 자세한 API 내용은 명령 참조ListDeviceProfiles의 섹션을 참조하세요. AWS CLI

### list-partner-accounts

다음 코드 예시에서는 list-partner-accounts을 사용하는 방법을 보여 줍니다.

AWS CLI

파트너 계정을 나열하려면

다음 list-partner-accounts 예제에서는 계정과 연결된 사용 가능한 파트너 AWS 계정을 나열합니다.

```
aws iotwireless list-partner-accounts
```

출력:

```
{
    "Sidewalk": [
        {
            "AmazonId": "78965678771228",
            "Fingerprint":
 "bd96d8ef66dbfd2160eb60e156849e82ad7018b8b73c1ba0b4fc65c32498ee35"
        },
        {
            "AmazonId": "89656787651228",
            "Fingerprint":
 "bc5e99e151c07be14be7e6603e4489c53f858b271213a36ebe3370777ba06e9b"
        }
    ]
}
```

자세한 내용은 AWS IoT 개발자 안내서의 IoT Core용 Amazon Sidewalk 통합AWS 을 참조하세요 IoT.

- 자세한 API 내용은 명령 참조ListPartnerAccounts의 섹션을 참조하세요. AWS CLI

### `list-service-profiles`

다음 코드 예시에서는 `list-service-profiles`을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스 프로파일을 나열하려면

다음 `list-service-profiles` 예제에서는 AWS 계정에 등록된 사용 가능한 서비스 프로파일을 나열합니다.

```
aws iotwireless list-service-profiles
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 프로필 추가 LoRaWANAWS 를 참조하세요 IoT.

- 자세한 API 내용은 명령 참조ListServiceProfiles의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 할당된 태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 무선 대상 리소스에 할당된 태그를 나열합니다.

```
aws iotwireless list-tags-for-resource \
    --resource-arn "arn:aws:iotwireless:us-east-1:123456789012:Destination/
IoTWirelessDestination"
```

출력:

```
{
    "Tags": [
        {
            "Value": "MyValue",
            "Key": "MyTag"
        }
    ]
}
```

자세한 내용은 AWS IoT 개발자 안내서의 리소스에 대한 LoRaWAN IoT Core 설명을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### `list-wireless-devices`

다음 코드 예시에서는 `list-wireless-devices`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 무선 디바이스를 나열하려면

다음 `list-wireless-devices` 예제에서는 AWS 계정에 등록된 사용 가능한 무선 디바이스를 나열합니다.

```
aws iotwireless list-wireless-devices
```

출력:

```
{
    "WirelessDeviceList": [
        {
            "Name": "myLoRaWANDevice",
            "DestinationName": "IoTWirelessDestination",
            "Id": "1ffd32c8-8130-4194-96df-622f072a315f",
            "Type": "LoRaWAN",
            "LoRaWAN": {
                "DevEui": "ac12efc654d23fc2"
            },
            "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:WirelessDevice/1ffd32c8-8130-4194-96df-622f072a315f"
        }
    ]
}
```

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 디바이스 및 게이트웨이 연결을 LoRaWAN AWS 참조하세요 IoT.

- 자세한 API 내용은 명령 참조ListWirelessDevices의 섹션을 참조하세요. AWS CLI

### `list-wireless-gateway-task-definitions`

다음 코드 예시에서는 `list-wireless-gateway-task-definitions`을 사용하는 방법을 보여 줍니다.

AWS CLI

무선 게이트웨이 작업 정의를 나열하려면

다음 `list-wireless-gateway-task-definitions` 예제에서는 AWS 계정에 등록된 사용 가능한 무선 게이트웨이 작업 정의를 나열합니다.

```
aws iotwireless list-wireless-gateway-task-definitions
```

출력:

```
{
    "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"
                }
            }
        }
    ]
}
```

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 디바이스 및 게이트웨이 연결을 LoRaWAN AWS 참조하세요 IoT.

- 자세한 API 내용은 명령 참조ListWirelessGatewayTaskDefinitions의 섹션을 참조하세요. AWS CLI

### `list-wireless-gateways`

다음 코드 예시에서는 `list-wireless-gateways`을 사용하는 방법을 보여 줍니다.

AWS CLI

무선 게이트웨이를 나열하려면

다음 `list-wireless-gateways` 예제에서는 AWS 계정에서 사용 가능한 무선 게이트웨이를 나열합니다.

```
aws iotwireless list-wireless-gateways
```

출력:

```
{
    "WirelessGatewayList": [
        {
            "Description": "My first LoRaWAN gateway",
            "LoRaWAN": {
                "RfRegion": "US915",
                "GatewayEui": "dac632ebc01d23e4"
            },
            "Id": "3039b406-5cc9-4307-925b-9948c63da25b",
            "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:WirelessGateway/3039b406-5cc9-4307-925b-9948c63da25b",
            "Name": "myFirstLoRaWANGateway"
        },
        {
            "Description": "My second LoRaWAN gateway",
            "LoRaWAN": {
                "RfRegion": "US915",
                "GatewayEui": "cda123fffe92ecd2"
            },
            "Id": "3285bdc7-5a12-4991-84ed-dadca65e342e",
            "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:WirelessGateway/3285bdc7-5a12-4991-84ed-dadca65e342e",
            "Name": "mySecondLoRaWANGateway"
        }
    ]
}
```

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 디바이스 및 게이트웨이 연결을 LoRaWAN AWS 참조하세요 IoT.

- 자세한 API 내용은 명령 참조ListWirelessGateways의 섹션을 참조하세요. AWS CLI

### send-data-to-wireless-device

다음 코드 예시에서는 send-data-to-wireless-device을 사용하는 방법을 보여 줍니다.

AWS CLI

무선 디바이스로 데이터를 보내려면

다음 send-data-to-wireless-device 예제에서는 복호화된 애플리케이션 데이터 프레임을 무선 디바이스로 보냅니다.

```
aws iotwireless send-data-to-wireless-device \
    --id "11aa5eae-2f56-4b8e-a023-b28d98494e49" \
    --transmit-mode "1" \
    --payload-data "SGVsbG8gVG8gRGV2c2lt" \
    --wireless-metadata LoRaWAN={FPort=1}
```

출력:

```
{
    MessageId: "6011dd36-0043d6eb-0072-0008"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 디바이스 및 게이트웨이 연결을 LoRaWAN AWS 참조하세요 IoT.

- 자세한 API 내용은 명령 참조SendDataToWirelessDevice의 섹션을 참조하세요. AWS CLI

### tag-resource

다음 코드 예시에서는 tag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스의 태그 키 및 값을 지정하려면

다음 tag-resource 예제에서는 키 MyTag 및 값 IoTWirelessDestination로 무선 대상에 태그를 지정합니다MyValue.

```
aws iotwireless tag-resource \
    --resource-arn "arn:aws:iotwireless:us-east-1:651419225604:Destination/
IoTWirelessDestination" \
```

```
    --tags Key="MyTag",Value="MyValue"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 리소스에 대한 LoRaWAN IoT Core 설명을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### test-wireless-device

다음 코드 예시에서는 test-wireless-device을 사용하는 방법을 보여 줍니다.

AWS CLI

무선 디바이스를 테스트하려면

다음 test-wireless-device 예제에서는 의 업링크 데이터를 지정된 ID를 가진 디바이스Hello로 전송합니다.

```
aws iotwireless test-wireless-device \
    --id "11aa5eae-2f56-4b8e-a023-b28d98494e49"
```

출력:

```
{
    Result: "Test succeeded. one message is sent with payload: hello"
}
```

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 디바이스 및 게이트웨이 연결을 LoRaWAN AWS 참조하세요 IoT.

- 자세한 API 내용은 명령 참조TestWirelessDevice의 섹션을 참조하세요. AWS CLI

### untag-resource

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에서 하나 이상의 태그를 제거하려면

다음 untag-resource 예제에서는 무선 대상 에서 태그MyTag와 해당 값을 제거합니다IoTWirelessDestination.

```
aws iotwireless untag-resource \
    --resource-arn "arn:aws:iotwireless:us-east-1:123456789012:Destination/
IoTWirelessDestination" \
    --tag-keys "MyTag"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 리소스에 대한 LoRaWAN IoT Core 설명을 참조하세요. AWS IoT

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-destination

다음 코드 예시에서는 update-destination을 사용하는 방법을 보여 줍니다.

AWS CLI

대상의 속성을 업데이트하려면

다음 update-destination 예제에서는 무선 대상의 설명 속성을 업데이트합니다.

```
aws iotwireless update-destination \
    --name "IoTWirelessDestination" \
    --description "Destination for messages processed using IoTWirelessRule"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 대상 추가 LoRaWANAWS 를 참조하세요 IoT.

- 자세한 API 내용은 명령 참조UpdateDestination의 섹션을 참조하세요. AWS CLI

### update-partner-account

다음 코드 예시에서는 update-partner-account을 사용하는 방법을 보여 줍니다.

AWS CLI

파트너 계정의 속성을 업데이트하려면

다음은 지정된 ID가 있는 계정AppServerPrivateKey의 를 update-partner-account 업데이트합니다.

```
aws iotwireless update-partner-account \
    --partner-account-id "78965678771228" \
    --partner-type "Sidewalk" \
    --sidewalk
 AppServerPrivateKey="f798ab4899346a88599180fee9e14fa1ada7b6df989425b7c6d2146dd6c815bb"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 IoT Core용 Amazon Sidewalk 통합AWS 을 참조하세요 IoT.

- 자세한 API 내용은 명령 참조UpdatePartnerAccount의 섹션을 참조하세요. AWS CLI

### update-wireless-device

다음 코드 예시에서는 update-wireless-device을 사용하는 방법을 보여 줍니다.

AWS CLI

무선 디바이스의 속성을 업데이트하려면

다음 update-wireless-device 예제에서는 AWS 계정에 등록된 무선 디바이스의 속성을 업데이트합니다.

```
aws iotwireless update-wireless-device \
    --id "1ffd32c8-8130-4194-96df-622f072a315f" \
    --destination-name IoTWirelessDestination2 \
    --description "Using my first LoRaWAN device"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 디바이스 및 게이트웨이 연결을 LoRaWAN AWS 참조하세요 IoT.

- 자세한 API 내용은 명령 참조UpdateWirelessDevice의 섹션을 참조하세요. AWS CLI

### update-wireless-gateway

다음 코드 예시에서는 update-wireless-gateway을 사용하는 방법을 보여 줍니다.

AWS CLI

무선 게이트웨이를 업데이트하려면

다음 update-wireless-gateway 예제에서는 무선 게이트웨이에 대한 설명을 업데이트합니다.

```
aws iotwireless update-wireless-gateway \
    --id "3285bdc7-5a12-4991-84ed-dadca65e342e" \
    --description "Using my LoRaWAN gateway"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS IoT 개발자 안내서의 용 IoT Core에 디바이스 및 게이트웨이 연결을 LoRaWAN AWS 참조하세요 IoT.

- 자세한 API 내용은 명령 참조UpdateWirelessGateway의 섹션을 참조하세요. AWS CLI

## 를 사용한 Amazon IVS 예제 AWS CLI

다음 코드 예제에서는 Amazon 와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다IVS.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### batch-get-channel

다음 코드 예시에서는 batch-get-channel을 사용하는 방법을 보여 줍니다.

AWS CLI

여러 채널에 대한 채널 구성 정보를 가져오려면

다음 batch-get-channel 예제에서는 지정된 채널에 대한 정보를 나열합니다.

```
aws ivs batch-get-channel \
    --arns arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \
        arn:aws:ivs:us-west-2:123456789012:channel/efghEFGHijkl
```

출력:

```
{
    "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":
 "AB1C2defGHijkLMNo3PqQRstUvwxyzaBCDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
            },
            "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":
 "BA1C2defGHijkLMNo3PqQRstUvwxyzaBCDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
            },
            "tags": {},
            "type": "STANDARD"
        }
    ]
}
```

자세한 내용은 IVS 지연 시간이 짧은 사용 설명서의 채널 생성을 참조하세요.

- 자세한 API 내용은 명령 참조BatchGetChannel의 섹션을 참조하세요. AWS CLI

### batch-get-stream-key

다음 코드 예시에서는 batch-get-stream-key을 사용하는 방법을 보여 줍니다.

AWS CLI

여러 스트림 키에 대한 정보를 가져오려면

다음 batch-get-stream-key 예제에서는 지정된 스트림 키에 대한 정보를 가져옵니다.

```
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
```

출력:

```
{
    "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": {}
        }
    ]
}
```

자세한 내용은 IVS 지연 시간이 짧은 사용 설명서의 채널 생성을 참조하세요.

- 자세한 API 내용은 명령 참조BatchGetStreamKey의 섹션을 참조하세요. AWS CLI

### batch-start-viewer-session-revocation

다음 코드 예시에서는 batch-start-viewer-session-revocation을 사용하는 방법을 보여 줍니다.

AWS CLI

여러 채널 및ARN 뷰어 ID 페어에 대한 뷰어 세션을 취소하려면

다음 batch-start-viewer-session-revocation 예제에서는 여러 채널 IDARN 페어와 뷰어 ID 페어에서 동시에 세션 취소를 수행합니다. 호출자에게 지정된 세션을 취소할 권한이 없는 경우 요청이 정상적으로 완료될 수 있지만 오류 필드에 값을 반환합니다.

```
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}]'
```

출력:

```
{
    "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",
        }
    ]
}
```

자세한 내용을 알아보려면 Amazon Interactive Video Service 사용 설명서의 프라이빗 채널 설정을 참조하세요.

- 자세한 API 내용은 명령 참조BatchStartViewerSessionRevocation의 섹션을 참조하세요. AWS CLI

### `create-channel`

다음 코드 예시에서는 `create-channel`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 녹음이 없는 채널을 생성하려면

다음 `create-channel` 예제에서는 스트리밍을 시작하기 위한 새 채널과 연결된 스트림 키를 생성합니다.

```
aws ivs create-channel \
    --name "test-channel" \
    --no-insecure-ingest
```

출력:

```
{
    "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": {}
    }
}
```

자세한 내용은 IVS 지연 시간이 짧은 사용 설명서의 채널 생성을 참조하세요.

예제 2: 에서 지정한 RecordingConfiguration 리소스를 사용하여 레코딩이 활성화된 채널을 생성하려면 ARN

다음 `create-channel` 예제에서는 스트리밍을 시작하기 위한 새 채널과 연결된 스트림 키를 생성하고 채널에 대한 레코딩을 설정합니다.

```
aws ivs create-channel \
    --name test-channel-with-recording \
    --insecure-ingest \
    --recording-configuration-arn "arn:aws:ivs:us-west-2:123456789012:recording-
configuration/ABCD12cdEFgh"
```

출력:

```
{
    "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": {}
    }
}
```

자세한 내용은 IVS 지연 시간이 짧은 사용 설명서의 Amazon S3에 기록을 참조하세요.

예제 3: 에서 지정한 재생 제한 정책으로 채널을 생성하려면 ARN

다음 `create-channel` 예제에서는 스트리밍을 시작하기 위한 새 채널과 연결된 스트림 키를 생성하고 채널에 대한 재생 제한 정책을 설정합니다.

```
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"
```

출력:

```
{
    "channel": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "name": "test-channel-with-playback-restriction-policy",
```

```
        "latencyMode": "LOW",
        "type": "STANDARD",
        "playbackRestrictionPolicyArn": "arn:aws:ivs:us-
west-2:123456789012:playback-restriction-policy/ABcdef34ghIJ",
        "recordingConfigurationArn": "",
        "srt": {
            "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
            "passphrase":
 "AB1C2edfGHijkLMNo3PqQRstUvwxyzaBCDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
        },
        "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
        "insecureIngest": true,
        "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
        "preset": "",
        "authorized": false,
        "tags": {},
        "type": "STANDARD"
    },
    "streamKey": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:stream-key/abcdABCDefgh",
        "value": "sk_us-west-2_abcdABCDefgh_567890abcdef",
        "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "tags": {}
    }
}
```

자세한 내용은 IVS 지연 시간이 짧은 사용 설명서의 원치 않는 콘텐츠 및 뷰어를 참조하세요.

- 자세한 API 내용은 명령 참조CreateChannel의 섹션을 참조하세요. AWS CLI

### create-playback-restriction-policy

다음 코드 예시에서는 create-playback-restriction-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

재생 제한 정책을 생성하려면

다음 create-playback-restriction-policy 예제에서는 새 재생 제한 정책을 생성합니다.

```
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
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 IVS 지연 시간이 짧은 사용 설명서의 원치 않는 콘텐츠 및 뷰어를 참조하세요.

- 자세한 API 내용은 명령 참조CreatePlaybackRestrictionPolicy의 섹션을 참조하세요. AWS CLI

### `create-recording-configuration`

다음 코드 예시에서는 `create-recording-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

RecordingConfiguration 리소스를 생성하려면

다음 `create-recording-configuration` 예제에서는 Amazon S3에 대한 레코딩을 활성화하는 RecordingConfiguration 리소스를 생성합니다.

```
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_RESOLU
 \
    --destination-configuration s3={bucketName=demo-recording-bucket}
```

출력:

```
{
    "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"
            ]
        }
    }
}
```

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 Amazon S3에 녹화를 참조하세요.

- 자세한 API 내용은 명령 참조CreateRecordingConfiguration의 섹션을 참조하세요. AWS CLI

### `create-stream-key`

다음 코드 예시에서는 `create-stream-key`을 사용하는 방법을 보여 줍니다.

AWS CLI

스트림 키를 생성하려면

다음 `create-stream-key` 예제에서는 지정된 ARN (Amazon 리소스 이름)에 대한 스트림 키를 생성합니다.

```
aws ivs create-stream-key \
    --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh
```

출력:

```
{
    "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": {}
    }
}
```

자세한 내용은 IVS 지연 시간이 짧은 사용 설명서의 채널 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateStreamKey의 섹션을 참조하세요. AWS CLI

### `delete-channel`

다음 코드 예시에서는 `delete-channel`을 사용하는 방법을 보여 줍니다.

AWS CLI

채널 및 관련 스트림 키를 삭제하려면

다음 delete-channel 예제에서는 ARN (Amazon 리소스 이름)이 지정된 채널을 삭제합니다.

```
aws ivs delete-channel \
    --arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IVS 지연 시간이 짧은 사용 설명서의 채널 생성을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteChannel의 섹션을 참조하세요. AWS CLI

### delete-playback-key-pair

다음 코드 예시에서는 delete-playback-key-pair을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 재생 키 페어를 삭제하려면

다음 delete-playback-key-pair 예제에서는 지정된 키 페어의 지문을 반환합니다.

```
aws ivs delete-playback-key-pair \
    --arn arn:aws:ivs:us-west-2:123456789012:playback-key/abcd1234efgh
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용을 알아보려면 Amazon Interactive Video Service 사용 설명서의 프라이빗 채널 설정을 참조하세요.

- 자세한 API 내용은 명령 참조DeletePlaybackKeyPair의 섹션을 참조하세요. AWS CLI

### delete-playback-restriction-policy

다음 코드 예시에서는 delete-playback-restriction-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

재생 제한 정책을 삭제하려면

다음 delete-playback-restriction-policy 예제에서는 지정된 정책ARN(Amazon 리소스 이름)을 사용하여 재생 제한 정책을 삭제합니다.

```
aws ivs delete-playback-restriction-policy \
    --arn "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/
ABcdef34ghIJ"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IVS 지연 시간이 짧은 사용 설명서의 원치 않는 콘텐츠 및 뷰어를 참조하세요.

- 자세한 API 내용은 명령 참조DeletePlaybackRestrictionPolicy의 섹션을 참조하세요. AWS CLI

### `delete-recording-configuration`

다음 코드 예시에서는 `delete-recording-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 지정한 RecordingConfiguration 리소스를 삭제하려면 ARN

다음 `delete-recording-configuration` 예제에서는 지정된 를 사용하여 RecordingConfiguration 리소스를 삭제합니다ARN.

```
aws ivs delete-recording-configuration \
    --arn "arn:aws:ivs:us-west-2:123456789012:recording-configuration/ABcdef34ghIJ"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 Amazon S3에 녹화를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteRecordingConfiguration의 섹션을 참조하세요. AWS CLI

### `delete-stream-key`

다음 코드 예시에서는 `delete-stream-key`을 사용하는 방법을 보여 줍니다.

AWS CLI

스트림 키를 삭제하려면

다음 `delete-stream-key` 예제에서는 지정된 ARN (Amazon 리소스 이름)의 스트림 키를 삭제하므로 더 이상 스트리밍에 사용할 수 없습니다.

```
aws ivs delete-stream-key \
    --arn arn:aws:ivs:us-west-2:123456789012:stream-key/g1H2I3j4k5L6
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IVS 지연 시간이 짧은 사용 설명서의 채널 생성을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteStreamKey의 섹션을 참조하세요. AWS CLI

### get-channel

다음 코드 예시에서는 get-channel을 사용하는 방법을 보여 줍니다.

AWS CLI

채널의 구성 정보를 가져오려면

다음 get-channel 예제에서는 지정된 채널ARN(Amazon 리소스 이름)에 대한 채널 구성을 가져옵니다.

```
aws ivs get-channel \
    --arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh
```

출력:

```
{
    "channel": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "name": "channel-1",
        "latencyMode": "LOW",
        "type": "STANDARD",
        "playbackRestrictionPolicyArn": "",
        "preset": "",
        "recordingConfigurationArn": "arn:aws:ivs:us-west-2:123456789012:recording-
configuration/ABCD12cdEFgh",
        "srt": {
            "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
            "passphrase":
 "AB1C2defGHijkLMNo3PqQRstUvwxyzaBCDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
        },
        "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
        "insecureIngest": false,
```

```
        "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
        "tags": {}
    }
}
```

자세한 내용은 IVS 지연 시간이 짧은 사용 설명서의 채널 생성을 참조하세요.

- 자세한 API 내용은 명령 참조GetChannel의 섹션을 참조하세요. AWS CLI

### get-playback-key-pair

다음 코드 예시에서는 get-playback-key-pair을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 재생 키 페어를 가져오려면

다음 get-playback-key-pair 예제에서는 지정된 키 페어의 지문을 반환합니다.

```
aws ivs get-playback-key-pair \
    --arn arn:aws:ivs:us-west-2:123456789012:playback-key/abcd1234efgh
```

출력:

```
{
    "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": {}
    }
}
```

자세한 내용을 알아보려면 Amazon Interactive Video Service 사용 설명서의 프라이빗 채널 설정을 참조하세요.

- 자세한 API 내용은 명령 참조GetPlaybackKeyPair의 섹션을 참조하세요. AWS CLI

### get-playback-restriction-policy

다음 코드 예시에서는 get-playback-restriction-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

재생 제한 정책의 구성 정보를 가져오려면

다음 `get-playback-restriction-policy` 예제에서는 지정된 정책ARN(Amazon 리소스 이름)을 사용하여 재생 제한 정책 구성을 가져옵니다.

```
aws ivs get-playback-restriction-policy \
    --arn "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/
ABcdef34ghIJ"
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 IVS 지연 시간이 짧은 사용 설명서의 원치 않는 콘텐츠 및 뷰어를 참조하세요.

- 자세한 API 내용은 명령 참조GetPlaybackRestrictionPolicy의 섹션을 참조하세요. AWS CLI

### `get-recording-configuration`

다음 코드 예시에서는 `get-recording-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

RecordingConfiguration 리소스에 대한 정보를 가져오려면

다음 `get-recording-configuration` 예제에서는 지정된 의 RecordingConfiguration 리소스에 대한 정보를 가져옵니다ARN.

```
aws ivs get-recording-configuration \
    --arn "arn:aws:ivs:us-west-2:123456789012:recording-configuration/ABcdef34ghIJ"
```

출력:

```
{
    "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"
            ]
        }
    }
```

```
}
```

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 Amazon S3에 레코드를 참조하세요.

- 자세한 API 내용은 명령 참조GetRecordingConfiguration의 섹션을 참조하세요. AWS CLI

### get-stream-key

다음 코드 예시에서는 get-stream-key을 사용하는 방법을 보여 줍니다.

AWS CLI

스트림에 대한 정보를 가져오려면

다음 get-stream-key 예제에서는 지정된 스트림 키에 대한 정보를 가져옵니다.

```
aws ivs get-stream-key \
    --arn arn:aws:ivs:us-west-2:123456789012:stream-key/skSKABCDefgh --region=us-
west-2
```

출력:

```
{
    "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": {}
    }
}
```

자세한 내용은 IVS 지연 시간이 짧은 사용 설명서의 채널 생성을 참조하세요.

- 자세한 API 내용은 명령 참조GetStreamKey의 섹션을 참조하세요. AWS CLI

### get-stream-session

다음 코드 예시에서는 get-stream-session을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 스트림의 메타데이터를 가져오려면

다음 get-stream-session 예제에서는 지정된 채널ARN(Amazon 리소스 이름) 및 지정된 스트림에 대한 메타데이터 구성을 가져옵니다. streamId 가 제공되지 않으면 채널의 최신 스트림이 선택됩니다.

```
aws ivs get-stream-session \
    --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \
    --stream-id "mystream"
```

출력:

```
{
    "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"
            }
        ]
    }
}
```

자세한 내용은 IVS 지연 시간이 짧은 사용 설명서의 채널 생성을 참조하세요.

- 자세한 API 내용은 명령 참조GetStreamSession의 섹션을 참조하세요. AWS CLI

### get-stream

다음 코드 예시에서는 get-stream을 사용하는 방법을 보여 줍니다.

AWS CLI

스트림에 대한 정보를 가져오려면

다음 get-stream 예제에서는 지정된 채널의 스트림에 대한 정보를 가져옵니다.

```
aws ivs get-stream \
    --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh
```

출력:

```
{
    "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
```

```
    }
}
```

자세한 내용은 IVS 지연 시간이 짧은 사용 설명서의 채널 생성을 참조하세요.

- 자세한 API 내용은 명령 참조GetStream의 섹션을 참조하세요. AWS CLI

### `import-playback-key-pair`

다음 코드 예시에서는 `import-playback-key-pair`을 사용하는 방법을 보여 줍니다.

AWS CLI

새 키 페어의 퍼블릭 부분을 가져오려면

다음 `import-playback-key-pair` 예제에서는 지정된 퍼블릭 키( PEM 형식의 문자열로 지정됨)를 가져오고 새 키 페어의 arn 및 지문을 반환합니다.

```
aws ivs import-playback-key-pair \
    --name "my-playback-key" \
    --public-key-material "G1lbnQxOTA3BgNVBAMMMFdoeSBhcmUgeW91IGRl..."
```

출력:

```
{
    "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": {}
    }
}
```

자세한 내용을 알아보려면 Amazon Interactive Video Service 사용 설명서의 프라이빗 채널 설정을 참조하세요.

- 자세한 API 내용은 명령 참조ImportPlaybackKeyPair의 섹션을 참조하세요. AWS CLI

### `list-channels`

다음 코드 예시에서는 `list-channels`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 모든 채널에 대한 요약 정보를 가져오려면

다음 `list-channels` 예제에서는 AWS 계정의 모든 채널을 나열합니다.

```
aws ivs list-channels
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 IVS 지연 시간이 짧은 사용 설명서의 채널 생성을 참조하세요.

예제 2: 모든 채널에 대한 요약 정보를 가져오려면 지정된 로 필터링합니다. RecordingConfiguration ARN

다음 `list-channels` 예제에서는 지정된 와 연결된 AWS 계정의 모든 채널을 나열합니다 RecordingConfiguration ARN.

```
aws ivs list-channels \
    --filter-by-recording-configuration-arn "arn:aws:ivs:us-
west-2:123456789012:recording-configuration/ABCD12cdEFgh"
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 IVS 지연 시간이 짧은 사용 설명서의 Amazon S3에 레코드를 참조하세요.

예제 3: 모든 채널에 대한 요약 정보를 가져오려면 지정된 로 필터링합니다. PlaybackRestrictionPolicy ARN

다음 `list-channels` 예제에서는 지정된 와 연결된 AWS 계정의 모든 채널을 나열합니다 PlaybackRestrictionPolicy ARN.

```
aws ivs list-channels \
    --filter-by-playback-restriction-policy-arn "arn:aws:ivs:us-
west-2:123456789012:playback-restriction-policy/ABcdef34ghIJ"
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 IVS 지연 시간이 짧은 사용 설명서의 원치 않는 콘텐츠 및 뷰어를 참조하세요.

- 자세한 API 내용은 명령 참조ListChannels의 섹션을 참조하세요. AWS CLI

### `list-playback-key-pairs`

다음 코드 예시에서는 `list-playback-key-pairs`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 재생 키 페어에 대한 요약 정보를 가져오려면

다음 `list-playback-key-pairs` 예제에서는 모든 키 페어에 대한 정보를 반환합니다.

```
aws ivs list-playback-key-pairs
```

출력:

```
{
    "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/ijkl5678mnop",
            "name": "test-key-1",
            "tags": {}
        }
    ]
}
```

자세한 내용을 알아보려면 Amazon Interactive Video Service 사용 설명서의 프라이빗 채널 설정을 참조하세요.

- 자세한 API 내용은 명령 참조ListPlaybackKeyPairs의 섹션을 참조하세요. AWS CLI

### `list-playback-restriction-policies`

다음 코드 예시에서는 `list-playback-restriction-policies`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 재생 제한 정책에 대한 요약 정보를 가져오려면

다음 `list-playback-restriction-policies` 예제에서는 AWS 계정에 대한 모든 재생 제한 정책을 나열합니다.

```
aws ivs list-playback-restriction-policies
```

출력:

```
{
    "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"
            }
        }
    ]
}
```

자세한 내용은 IVS 지연 시간이 짧은 사용 설명서의 원치 않는 콘텐츠 및 뷰어를 참조하세요.

- 자세한 API 내용은 명령 참조ListPlaybackRestrictionPolicies의 섹션을 참조하세요. AWS CLI

### `list-recording-configurations`

다음 코드 예시에서는 `list-recording-configurations`을 사용하는 방법을 보여 줍니다.

AWS CLI

이 계정에서 생성된 모든 RecordingConfiguration 리소스를 나열하려면

다음 `list-recording-configurations` 예제에서는 계정의 모든 RecordingConfiguration 리소스에 대한 정보를 가져옵니다.

```
aws ivs list-recording-configurations
```

출력:

```
{
    "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": {}
        }
    ]
}
```

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 Amazon S3에 레코드를 참조하세요.

- 자세한 API 내용은 명령 참조ListRecordingConfigurations의 섹션을 참조하세요. AWS CLI

### `list-stream-keys`

다음 코드 예시에서는 `list-stream-keys`을 사용하는 방법을 보여 줍니다.

AWS CLI

스트림 키 목록을 가져오려면

다음 `list-stream-keys` 예제에서는 지정된 ARN (Amazon 리소스 이름)의 모든 스트림 키를 나열합니다.

```
aws ivs list-stream-keys \
    --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh
```

출력:

```
{
    "streamKeys": [
        {
            "arn": "arn:aws:ivs:us-west-2:123456789012:stream-key/abcdABCDefgh",
            "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
```

```
            "tags": {}
        }
    ]
}
```

FFor 자세한 내용은 IVS 지연 시간이 짧은 사용 설명서의 채널 생성을 참조하세요.

- 자세한 API 내용은 명령 참조ListStreamKeys의 섹션을 참조하세요. AWS CLI

### `list-stream-sessions`

다음 코드 예시에서는 `list-stream-sessions`을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 AWS 리전의 지정된 채널에 대한 현재 및 이전 스트림의 요약을 가져오려면

다음 `list-stream-sessions` 예제에서는 지정된 채널ARN(Amazon 리소스 이름)의 스트림에 대한 요약 정보를 보고합니다.

```
aws ivs list-stream-sessions \
    --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \
    --max-results 25 \
    --next-token ""
```

출력:

```
{
    "nextToken": "set-2",
    "streamSessions": [
        {
            "startTime": 1641578182,
            "endTime": 1641579982,
            "hasErrorEvent": false,
            "streamId": "mystream"
        }
        ...
    ]
}
```

자세한 내용은 IVS 지연 시간이 짧은 사용 설명서의 채널 생성을 참조하세요.

- 자세한 API 내용은 명령 참조ListStreamSessions의 섹션을 참조하세요. AWS CLI

### list-streams

다음 코드 예시에서는 list-streams을 사용하는 방법을 보여 줍니다.

AWS CLI

라이브 스트림 목록과 상태를 가져오려면

다음 list-streams 예제에서는 AWS 계정의 모든 라이브 스트림을 나열합니다.

```
aws ivs list-streams
```

출력:

```
{
   "streams": [
        {
            "channelArn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
            "state": "LIVE",
            "health": "HEALTHY",
            "streamId": "st-ABCDEfghij01234KLMN5678",
            "viewerCount": 1
        }
    ]
}
```

자세한 내용은 IVS 지연 시간이 짧은 사용 설명서의 채널 생성을 참조하세요.

- 자세한 API 내용은 명령 참조ListStreams의 섹션을 참조하세요. AWS CLI

### list-tags-for-resource

다음 코드 예시에서는 list-tags-for-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 리소스의 모든 태그를 나열하려면(예: 채널, 스트림 키)

다음 list-tags-for-resource 예제에서는 지정된 리소스ARN(Amazon 리소스 이름)에 대한 모든 태그를 나열합니다.

```
aws ivs list-tags-for-resource \
```

```
    --resource-arn arn:aws:ivs:us-west-2:12345689012:channel/abcdABCDefgh
```

출력:

```
{
    "tags":
    {
        "key1": "value1",
        "key2": "value2"
    }
}
```

자세한 내용은 Amazon Interactive Video Service API 참조의 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### put-metadata

다음 코드 예시에서는 put-metadata을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 채널의 활성 스트림에 메타데이터를 삽입하려면

다음 put-metadata 예제에서는 지정된 메타데이터를 지정된 채널의 스트림에 삽입합니다.

```
aws ivs put-metadata \
    --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \
    --metadata '{"my": "metadata"}'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IVS 지연 시간이 짧은 사용 설명서의 채널 생성을 참조하세요.

- 자세한 API 내용은 명령 참조PutMetadata의 섹션을 참조하세요. AWS CLI

### start-viewer-session-revocation

다음 코드 예시에서는 start-viewer-session-revocation을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 다중 채널 및ARN 뷰어 ID 페어에 대한 뷰어 세션을 취소하려면

다음 start-viewer-session-revocation 예제에서는 지정된 채널 ARN 및 뷰어 ID와 연결된 뷰어 세션을 지정된 세션 버전 번호까지 취소하는 프로세스를 시작합니다. 버전이 제공되지 않는 경우 기본값은 0입니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용을 알아보려면 Amazon Interactive Video Service 사용 설명서의 프라이빗 채널 설정을 참조하세요.

- 자세한 API 내용은 명령 참조StartViewerSessionRevocation의 섹션을 참조하세요. AWS CLI

### stop-stream

다음 코드 예시에서는 stop-stream을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 스트림을 중지하려면

다음 stop-stream 예제는 지정된 채널에서 스트림을 중지합니다.

```
aws ivs stop-stream \
    --channel-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 IVS 지연 시간이 짧은 사용 설명서의 채널 생성을 참조하세요.

- 자세한 API 내용은 명령 참조StopStream의 섹션을 참조하세요. AWS CLI

### tag-resource

다음 코드 예시에서는 tag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 리소스에 대한 태그를 추가하거나 업데이트하려면(예: 채널, 스트림 키)

다음 `tag-resource` 예제에서는 지정된 리소스ARN(Amazon 리소스 이름)에 대한 태그를 추가하거나 업데이트합니다.

```
aws ivs tag-resource \
    --resource-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \
    --tags "tagkey1=tagvalue1, tagkey2=tagvalue2"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Interactive Video Service API 참조의 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### `untag-resource`

다음 코드 예시에서는 `untag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 리소스의 태그를 제거하려면(예: 채널, 스트림 키)

다음 `untag-resource` 예제에서는 지정된 리소스ARN(Amazon 리소스 이름)에 대해 지정된 태그를 제거합니다.

```
aws ivs untag-resource \
    --resource-arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \
    --tag-keys "tagkey1, tagkey2"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Interactive Video Service API 참조의 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### `update-channel`

다음 코드 예시에서는 `update-channel`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 채널의 구성 정보를 업데이트하려면

다음 update-channel 예제에서는 지정된 채널의 채널 구성을 업데이트ARN하여 채널 이름을 변경합니다. 이는 이 채널의 진행 중인 스트림에는 영향을 주지 않습니다. 변경 사항을 적용하려면 스트림을 중지했다가 다시 시작해야 합니다.

```
aws ivs update-channel \
    --arn arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh \
    --name "channel-1" \
    --insecure-ingest
```

출력:

```
{
    "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": {}
}
```

자세한 내용은 IVS 지연 시간이 짧은 사용 설명서의 채널 생성을 참조하세요.

예제 2: 채널의 구성을 업데이트하여 레코딩을 활성화하는 방법

다음 update-channel 예제에서는 지정된 채널에 대한 채널 구성을 업데이트ARN하여 레코딩을 활성화합니다. 이는 이 채널의 진행 중인 스트림에는 영향을 주지 않습니다. 변경 사항을 적용하려면 스트림을 중지했다가 다시 시작해야 합니다.

```
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"
```

출력:

```
{
    "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": {}
    }
}
```

자세한 내용은 IVS 지연 시간이 짧은 사용 설명서의 Amazon S3에 레코드를 참조하세요.

예제 3: 채널의 구성을 업데이트하여 레코딩을 비활성화하는 방법

다음 update-channel 예제에서는 지정된 채널에 대한 채널 구성을 업데이트ARN하여 레코딩을 비활성화합니다. 이는 이 채널의 진행 중인 스트림에는 영향을 주지 않습니다. 변경 사항을 적용하려면 스트림을 중지했다가 다시 시작해야 합니다.

```
aws ivs update-channel \
    --arn "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh" \
    --recording-configuration-arn ""
```

출력:

```
{
    "channel": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "name": "test-channel-with-recording",
        "latencyMode": "LOW",
        "type": "STANDARD",
        "playbackRestrictionPolicyArn": "",
        "recordingConfigurationArn": "",
        "srt": {
            "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
            "passphrase":
 "AB1C2edfGHijkLMNo3PqQRstUvwxyzaBCDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
        },
        "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
        "insecureIngest": false,
        "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
        "preset": "",
        "authorized": false,
        "tags": {}
    }
}
```

자세한 내용은 IVS 지연 시간이 짧은 사용 설명서의 Amazon S3에 레코드를 참조하세요.

예제 4: 재생 제한을 활성화하도록 채널의 구성을 업데이트하려면

다음 update-channel 예제에서는 지정된 채널에 대한 채널 구성을 업데이트ARN하여 재생 제한 정책을 적용합니다. 이는 이 채널의 진행 중인 스트림에는 영향을 주지 않습니다. 변경 사항을 적용하려면 스트림을 중지했다가 다시 시작해야 합니다.

```
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"
```

출력:

```
{
    "channel": {
```

```
        "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "name": "test-channel-with-playback-restriction-policy",
        "latencyMode": "LOW",
        "type": "STANDARD",
        "playbackRestrictionPolicyArn": "arn:aws:ivs:us-
west-2:123456789012:playback-restriction-policy/ABcdef34ghIJ",
        "recordingConfigurationArn": "",
        "srt": {
            "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
            "passphrase":
 "AB1C2defGHijkLMNo3PqQRstUvwxyzaCBDEfghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
        },
        "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
        "insecureIngest": false,
        "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
        "preset": "",
        "authorized": false,
        "tags": {}
    }
}
```

자세한 내용은 IVS 지연 시간이 짧은 사용 설명서의 원치 않는 콘텐츠 및 뷰어를 참조하세요.

예제 5: 채널 구성을 업데이트하여 재생 제한을 비활성화하는 방법

다음 update-channel 예제에서는 지정된 채널의 채널 구성을 업데이트ARN하여 재생 제한을 비활성화합니다. 이는 이 채널의 진행 중인 스트림에는 영향을 주지 않습니다. 변경 사항을 적용하려면 스트림을 중지했다가 다시 시작해야 합니다.

```
aws ivs update-channel \
    --arn "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh" \
    --playback-restriction-policy-arn ""
```

출력:

```
{
    "channel": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:channel/abcdABCDefgh",
        "name": "test-channel-with-playback-restriction-policy",
        "latencyMode": "LOW",
        "type": "STANDARD",
        "playbackRestrictionPolicyArn": "",
```

```
        "recordingConfigurationArn": "",
        "srt": {
            "endpoint": "a1b2c3d4e5f6.srt.live-video.net",
            "passphrase":
 "AB1C2defGHijkLMNo3PqQRstUvwxyzaBCDeFghh4ijklMN5opqrStuVWxyzAbCDEfghIJ"
        },
        "ingestEndpoint": "a1b2c3d4e5f6.global-contribute.live-video.net",
        "insecureIngest": false,
        "playbackUrl": "https://a1b2c3d4e5f6.us-west-2.playback.live-video.net/api/
video/v1/us-west-2.123456789012.channel.abcdEFGH.m3u8",
        "preset": "",
        "authorized": false,
        "tags": {}
    }
}
```

자세한 내용은 IVS 지연 시간이 짧은 사용 설명서의 원치 않는 콘텐츠 및 뷰어를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateChannel의 섹션을 참조하세요. AWS CLI

### update-playback-restriction-policy

다음 코드 예시에서는 update-playback-restriction-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

재생 제한 정책을 업데이트하려면

다음 update-playback-restriction-policy 예제에서는 재생 제한 정책을 지정된 정책으로 업데이트ARN하여 엄격한 오리진 적용을 비활성화합니다. 이는 연결된 채널의 진행 중인 스트림에는 영향을 주지 않습니다. 변경 사항을 적용하려면 스트림을 중지했다가 다시 시작해야 합니다.

```
aws ivs update-playback-restriction-policy \
    --arn "arn:aws:ivs:us-west-2:123456789012:playback-restriction-policy/
ABcdef34ghIJ" \
    --no-enable-strict-origin-enforcement
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 IVS 지연 시간이 짧은 사용 설명서의 원치 않는 콘텐츠 및 뷰어를 참조하세요.

- 자세한 API 내용은 명령 참조UpdatePlaybackRestrictionPolicy의 섹션을 참조하세요. AWS CLI

## 를 사용한 Amazon IVS Chat 예제 AWS CLI

다음 코드 예제에서는 Amazon IVS Chat AWS Command Line Interface 과 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `create-chat-token`

다음 코드 예시에서는 `create-chat-token`을 사용하는 방법을 보여 줍니다.

AWS CLI

채팅 토큰을 생성하려면

다음 `create-chat-token` 예제에서는 룸에 대한 개별 WebSocket 연결을 설정하는 데 사용되는 암호화된 채팅 토큰을 생성합니다. 토큰은 1분 동안 유효하며 토큰으로 설정된 연결(세션)은 지정된 기간 동안 유효합니다.

```
aws ivschat create-chat-token \
    --roomIdentifier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6", \
    --userId" "11231234" \
    --capabilities "SEND_MESSAGE", \
    --sessionDurationInMinutes" 30
```

출력:

```
{
    "token": "ACEGmnoq#1rstu2...BDFH3vxwy!4hlm!#5",
    "sessionExpirationTime": "2022-03-16T04:44:09+00:00"
    "state": "CREATING",
    "tokenExpirationTime": "2022-03-16T03:45:09+00:00"
}
```

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 3단계: 채팅 클라이언트 인증 및 권한 부여를 참조하세요.

- 자세한 API 내용은 명령 참조CreateChatToken의 섹션을 참조하세요. AWS CLI

### `create-logging-configuration`

다음 코드 예시에서는 `create-logging-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

채팅 LoggingConfiguration 리소스를 생성하려면

다음 `create-logging-configuration` 예제에서는 클라이언트가 전송된 메시지를 저장하고 기록할 수 있는 LoggingConfiguration 리소스를 생성합니다.

```
aws ivschat create-logging-configuration \
```

```
    --destination-configuration s3={bucketName=demo-logging-bucket} \
    --name "test-logging-config" \
    --tags "key1=value1, key2=value2"
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 Amazon IVS Chat 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조CreateLoggingConfiguration의 섹션을 참조하세요. AWS CLI

### create-room

다음 코드 예시에서는 create-room을 사용하는 방법을 보여 줍니다.

AWS CLI

방을 생성하려면

다음 create-room 예제에서는 새 방을 생성합니다.

```
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
```

출력:

```
{
    "arn": "arn:aws:ivschat:us-west-2:12345689012: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"
}
```

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 2단계: 채팅룸 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateRoom의 섹션을 참조하세요. AWS CLI

### delete-logging-configuration

다음 코드 예시에서는 delete-logging-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

채팅 LoggingConfiguration 리소스를 삭제하려면

다음 delete-logging-configuration 예제에서는 지정된 에 대한 LoggingConfiguration 리소스를 삭제합니다ARN.

```
aws ivschat delete-logging-configuration \
    --identifier "arn:aws:ivschat:us-west-2:123456789012:logging-configuration/
ABcdef34ghIJ"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 Amazon IVS Chat 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteLoggingConfiguration의 섹션을 참조하세요. AWS CLI

### delete-message

다음 코드 예시에서는 delete-message을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 방에서 메시지를 삭제하려면

다음 delete-message 예제에서는 지정된 방으로 짝수를 보내 클라이언트가 지정된 메시지를 삭제하도록 지시합니다. 즉, 뷰에서 렌더링을 취소하고 클라이언트의 채팅 기록에서 삭제합니다.

```
aws ivschat delete-message \
    --roomIdentifier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6" \
    --id "ABC123def456" \
    --reason "Message contains profanity"
```

출력:

```
{
    "id": "12345689012"
}
```

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 Amazon IVS Chat 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteMessage의 섹션을 참조하세요. AWS CLI

### delete-room

다음 코드 예시에서는 delete-room을 사용하는 방법을 보여 줍니다.

AWS CLI

방을 삭제하려면

다음 delete-room 예제에서는 지정된 방을 삭제합니다. 연결된 클라이언트가 연결 해제되었습니다. 성공하면 빈 응답 본문과 함께 HTTP 204를 반환합니다.

```
aws ivschat delete-room \
    --identifier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 Amazon IVS Chat 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteRoom의 섹션을 참조하세요. AWS CLI

### `disconnect-user`

다음 코드 예시에서는 `disconnect-user`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자를 방에서 연결 해제하려면

다음 `disconnect-user` 예제에서는 지정된 사용자의 모든 연결을 지정된 방에서 분리합니다. 성공하면 응답 본문이 빈 상태에서 HTTP 200을 반환합니다.

```
aws ivschat disconnect-user \
    --roomIdentifier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6" \
    --userId "ABC123def456" \
    --reason "Violated terms of service"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 Amazon IVS Chat 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조DisconnectUser의 섹션을 참조하세요. AWS CLI

### `get-logging-configuration`

다음 코드 예시에서는 `get-logging-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

LoggingConfiguration 리소스에 대한 정보를 가져오려면

다음 `get-logging-configuration` 예제에서는 지정된 의 LoggingConfiguration 리소스에 대한 정보를 가져옵니다ARN.

```
aws ivschat get-logging-configuration \
```

```
    --identifier "arn:aws:ivschat:us-west-2:123456789012:logging-configuration/
ABcdef34ghIJ"
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 Amazon IVS Chat 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조GetLoggingConfiguration의 섹션을 참조하세요. AWS CLI

### get-room

다음 코드 예시에서는 get-room을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 방을 가져오려면

다음 get-room 예제에서는 지정된 방에 대한 정보를 가져옵니다.

```
aws ivschat get-room \
    --identifier "arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6"
```

출력:

```
{
```

```
    "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"
}
```

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 Amazon IVS Chat 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조GetRoom의 섹션을 참조하세요. AWS CLI

### `list-logging-configurations`

다음 코드 예시에서는 `list-logging-configurations`을 사용하는 방법을 보여 줍니다.

AWS CLI

API 요청이 처리되는 AWS 리전의 사용자에 대한 모든 로깅 구성에 대한 요약 정보를 가져오려면

다음 `list-logging-configurations` 예제에서는 API 요청이 처리되는 AWS 리전의 사용자에 대한 모든 LoggingConfiguration 리소스에 대한 정보를 나열합니다.

```
aws ivschat list-logging-configurations \
    --max-results 2 \
    --next-token ""
```

출력:

```
{
    "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"
        }
        ...
    ]
}
```

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 Amazon IVS Chat 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조ListLoggingConfigurations의 섹션을 참조하세요. AWS CLI

### `list-rooms`

다음 코드 예시에서는 `list-rooms`을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 리전의 모든 방에 대한 요약 정보를 가져오려면

다음 `list-rooms` 예제에서는 요청이 처리되는 AWS 리전의 모든 방에 대한 요약 정보를 가져옵니다. 결과는 내림차순으로 정렬됩니다updateTime.

```
aws ivschat list-rooms \
    --logging-configuration-identifier "arn:aws:ivschat:us-
west-2:123456789012:logging-configuration/ABcdef34ghIJ" \
    --max-results 10 \
    --next-token ""
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 Amazon IVS Chat 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조ListRooms의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 리소스의 모든 태그를 나열하려면(예: 룸)

다음 `list-tags-for-resource` 예제에서는 지정된 리소스ARN(Amazon 리소스 이름)에 대한 모든 태그를 나열합니다.

```
aws ivschat list-tags-for-resource \
    --resource-arn arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6
```

출력:

```
{
    "tags":
    {
        "key1": "value1",
        "key2": "value2"
    }
}
```

자세한 내용은 Amazon Interactive Video Service API 참조의 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### `send-event`

다음 코드 예시에서는 `send-event`을 사용하는 방법을 보여 줍니다.

AWS CLI

룸으로 이벤트를 보내려면

다음 `send-event` 예제에서는 지정된 이벤트를 지정된 방으로 보냅니다.

```
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."
```

출력:

```
{
    "id": "12345689012"
}
```

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 Amazon IVS Chat 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조SendEvent의 섹션을 참조하세요. AWS CLI

### `tag-resource`

다음 코드 예시에서는 `tag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 리소스에 대한 태그를 추가하거나 업데이트하려면(예: 저장 폴더)

다음 `tag-resource` 예제에서는 지정된 리소스ARN(Amazon 리소스 이름)에 대한 태그를 추가하거나 업데이트합니다. 성공하면 빈 응답 본문과 함께 HTTP 200을 반환합니다.

```
aws ivschat tag-resource \
    --resource-arn arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6 \
    --tags "tagkey1=tagkeyvalue1, tagkey2=tagkeyvalue2"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Interactive Video Service API 참조의 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### `untag-resource`

다음 코드 예시에서는 `untag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 리소스의 태그를 제거하려면(예: 저장 폴더)

다음 `untag-resource` 예제에서는 지정된 리소스ARN(Amazon 리소스 이름)에 대해 지정된 태그를 제거합니다. 성공하면 빈 응답 본문과 함께 HTTP 200을 반환합니다.

```
aws ivschat untag-resource \
    --resource-arn arn:aws:ivschat:us-west-2:12345689012:room/g1H2I3j4k5L6 \
    --tag-keys "tagkey1, tagkey2"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Interactive Video Service API 참조의 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### `update-logging-configuration`

다음 코드 예시에서는 `update-logging-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

방의 로깅 구성을 업데이트하려면

다음 `update-logging-configuration` 예제에서는 LoggingConfiguration 리소스를 지정된 데이터로 업데이트합니다.

```
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"
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 Amazon IVS Chat 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateLoggingConfiguration의 섹션을 참조하세요. AWS CLI

### update-room

다음 코드 예시에서는 update-room을 사용하는 방법을 보여 줍니다.

AWS CLI

방의 구성을 업데이트하려면

다음 update-room 예제에서는 지정된 방의 구성을 지정된 데이터로 업데이트합니다.

```
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
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 Amazon IVS Chat 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateRoom의 섹션을 참조하세요. AWS CLI

## 를 사용한 Amazon IVS 실시간 스트리밍 예제 AWS CLI

다음 코드 예제에서는 Amazon IVS Real-Time Streaming과 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### `create-encoder-configuration`

다음 코드 예시에서는 `create-encoder-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 인코더 구성을 생성하려면

다음 `create-encoder-configuration` 예제에서는 지정된 속성을 사용하여 구성 인코더 구성을 생성합니다.

```
aws ivs-realtime create-encoder-configuration \
    --name test-ec --video bitrate=3500000,framerate=30.0,height=1080,width=1920
```

출력:

```
{
    "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
        }
    }
}
```

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 Amazon IVS Stream에서 다중 호스트 활성화를 참조하세요.

- 자세한 API 내용은 명령 참조CreateEncoderConfiguration의 섹션을 참조하세요. AWS CLI

### `create-participant-token`

다음 코드 예시에서는 `create-participant-token`을 사용하는 방법을 보여 줍니다.

AWS CLI

스테이지 참가자 토큰을 생성하려면

다음 create-participant-token 예제에서는 지정된 단계에 대한 참가자 토크를 생성합니다.

```
aws ivs-realtime create-participant-token \
    --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
    --user-id bob
```

출력:

```
{
    "participantToken": {
        "expirationTime": "2023-03-07T09:47:43+00:00",
        "participantId": "ABCDEfghij01234KLMN6789",
        "token": "abcd1234defg5678"
    }
}
```

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 Amazon IVS Stream에서 다중 호스트 활성화를 참조하세요.

- 자세한 API 내용은 명령 참조CreateParticipantToken의 섹션을 참조하세요. AWS CLI

### create-stage

다음 코드 예시에서는 create-stage을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 스테이지 생성

다음 create-stage 예제에서는 지정된 사용자에 대한 스테이지 및 스테이지 참가자 토큰을 생성합니다.

```
aws ivs-realtime create-stage \
    --name stage1 \
    --participant-token-configurations userId=alice
```

출력:

```
{
    "participantTokens": [
        {
            "participantId": "ABCDEfghij01234KLMN5678",
            "token": "a1b2c3d4567890ab",
            "userId": "alice"
        }
    ],
    "stage": {
        "activeSessionId": "st-a1b2c3d4e5f6g",
        "arn": "arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh",
        "endpoints": {
            "events": "wss://global.events.live-video.net",
            "whip": "https://1a2b3c4d5e6f.global-bm.whip.live-video.net"
        },
        "name": "stage1",
        "tags": {}
    }
}
```

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 Amazon IVS Stream에서 다중 호스트 활성화를 참조하세요.

예제 2: 스테이지를 생성하고 개별 참가자 레코딩을 구성하려면

다음 `create-stage` 예제에서는 스테이지를 생성하고 개별 참가자 레코딩을 구성합니다.

```
aws ivs-realtime create-stage \
    --name stage1 \
    --auto-participant-recording-configuration '{"mediaTypes":
 ["AUDIO_VIDEO"],"storageConfigurationArn": "arn:aws:ivs:us-
west-2:123456789012:storage-configuration/abcdABCDefgh"}'
```

출력:

```
{
    "stage": {
        "activeSessionId": "st-a1b2c3d4e5f6g",
        "arn": "arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh",
        "autoParticipantRecordingConfiguration": {
             "mediaTypes": [
                   "AUDIO_VIDEO"
```

```
            ],
            "storageConfigurationArn": "arn:aws:ivs:us-west-2:123456789012:storage-
configuration/abcdABCDefgh",
        },
        "endpoints": {
            "events": "wss://global.events.live-video.net",
            "whip": "https://1a2b3c4d5e6f.global-bm.whip.live-video.net"
        },
        "name": "stage1",
        "tags": {}
    }
}
```

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 Amazon IVS Stream에서 다중 호스트 활성화를 참조하세요.

- 자세한 API 내용은 명령 참조CreateStage의 섹션을 참조하세요. AWS CLI

### create-storage-configuration

다음 코드 예시에서는 create-storage-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 스토리지 구성을 생성하려면

다음 create-storage-configuration 예제에서는 지정된 속성을 사용하여 구성 스토리지 구성을 생성합니다.

```
aws ivs-realtime create-storage-configuration \
    --name "test-sc" --s3 "bucketName=test-bucket-name"
```

출력:

```
{
    "storageConfiguration": {
        "arn": "arn:aws:ivs:ap-northeast-1:123456789012:storage-configuration/
ABabCDcdEFef",
        "name": "test-sc",
        "s3": {
            "bucketName": "test-bucket-name"
        },
```

```
        "tags": {}
    }
}
```

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 Amazon IVS Stream에서 다중 호스트 활성화를 참조하세요.

- 자세한 API 내용은 명령 참조CreateStorageConfiguration의 섹션을 참조하세요. AWS CLI

### `delete-encoder-configuration`

다음 코드 예시에서는 `delete-encoder-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 인코더 구성을 삭제하려면

다음은 지정된 ARN (Amazon 리소스 이름)에서 지정한 구성 인코더 구성을 `delete-encoder-configuration` 삭제합니다.

```
aws ivs-realtime delete-encoder-configuration \
    --arn "arn:aws:ivs:ap-northeast-1:123456789012:encoder-configuration/
ABabCDcdEFef"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 Amazon IVS Stream에서 다중 호스트 활성화를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteEncoderConfiguration의 섹션을 참조하세요. AWS CLI

### `delete-public-key`

다음 코드 예시에서는 `delete-public-key`을 사용하는 방법을 보여 줍니다.

AWS CLI

퍼블릭 키를 삭제하려면

다음은 지정된 퍼블릭 키를 `delete-public-key` 삭제합니다.

```
aws ivs-realtime delete-public-key \
```

```
    --arn arn:aws:ivs:us-west-2:123456789012:public-key/abcdABC1efg2
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon 실시간 스트리밍 사용 설명서의 참가자 토큰 배포를 참조하세요. IVS

- 자세한 API 내용은 명령 참조DeletePublicKey의 섹션을 참조하세요. AWS CLI

### delete-stage

다음 코드 예시에서는 delete-stage을 사용하는 방법을 보여 줍니다.

AWS CLI

스테이지를 삭제하려면

다음 delete-stage 예제에서는 지정된 단계를 삭제합니다.

```
aws ivs-realtime delete-stage \
    --arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 Amazon IVS Stream에서 다중 호스트 활성화를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteStage의 섹션을 참조하세요. AWS CLI

### delete-storage-configuration

다음 코드 예시에서는 delete-storage-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 스토리지 구성을 삭제하려면

다음은 지정된 ARN (Amazon 리소스 이름)에서 지정한 구성 스토리지 구성을 delete-storage-configuration 삭제합니다.

```
aws ivs-realtime delete-storage-configuration \
    --arn "arn:aws:ivs:ap-northeast-1:123456789012:storage-configuration/
ABabCDcdEFef"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 Amazon IVS Stream에서 다중 호스트 활성화를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteStorageConfiguration의 섹션을 참조하세요. AWS CLI

### `disconnect-participant`

다음 코드 예시에서는 `disconnect-participant`을 사용하는 방법을 보여 줍니다.

AWS CLI

스테이지 참가자 연결 해제

다음 `disconnect-participant` 예제에서는 지정된 참가자를 지정된 스테이지에서 연결 해제합니다.

```
aws ivs-realtime disconnect-participant \
    --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
    --participant-id ABCDEfghij01234KLMN5678
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 Amazon IVS Stream에서 다중 호스트 활성화를 참조하세요.

- 자세한 API 내용은 명령 참조DisconnectParticipant의 섹션을 참조하세요. AWS CLI

### `get-composition`

다음 코드 예시에서는 `get-composition`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 기본 레이아웃 설정으로 구성을 가져오려면

다음 `get-composition` 예제에서는 지정된 ARN (Amazon 리소스 이름)의 구성을 가져옵니다.

```
aws ivs-realtime get-composition \
    --arn "arn:aws:ivs:ap-northeast-1:123456789012:composition/abcdABCDefgh"
```

출력:

```
{
    "composition": {
        "arn": "arn:aws:ivs:ap-northeast-1:123456789012:composition/abcdABCDefgh",
        "destinations": [
            {
                "configuration": {
                    "channel": {
                        "channelArn": "arn:aws:ivs:ap-
northeast-1:123456789012:channel/abcABCdefDEg",
                        "encoderConfigurationArn": "arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"
                    },
                    "name": ""
                },
                "id": "AabBCcdDEefF",
                "startTime": "2023-10-16T23:26:00+00:00",
                "state": "ACTIVE"
            },
            {
                "configuration": {
                    "name": "",
                    "s3": {
                        "encoderConfigurationArns": [
                            "arn:aws:ivs:arn:aws:ivs:ap-
northeast-1:123456789012:encoder-configuration/ABabCDcdEFef"
                        ],
                        "recordingConfiguration": {
                            "format": "HLS"
                        },
                        "storageConfigurationArn": "arn:arn:aws:ivs:ap-
northeast-1:123456789012:storage-configuration/FefABabCDcdE"
                    }
                },
                "detail": {
                    "s3": {
                        "recordingPrefix": "aBcDeFgHhGfE/AbCdEfGhHgFe/GHFabcgefABC/
composite"
                    }
                },
                "id": "GHFabcgefABC",
                "startTime": "2023-10-16T23:26:00+00:00",
                "state": "STARTING"
```

```
            }
        ],
        "layout": {
            "grid": {
                "featuredParticipantAttribute": ""
                "gridGap": 2,
                "omitStoppedVideo": false,
                "videoAspectRatio": "VIDEO",
                "videoFillMode": ""                }
        },
        "stageArn": "arn:aws:ivs:ap-northeast-1:123456789012:stage/defgABCDabcd",
        "startTime": "2023-10-16T23:24:00+00:00",
        "state": "ACTIVE",
        "tags": {}
    }
}
```

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 복합 레코딩(실시간 스트리밍)을 참조하세요.

예제 2: PiP 레이아웃을 사용하여 구성을 가져오려면

다음 `get-composition` 예제에서는 PiP 레이아웃을 사용하는 지정된 ARN (Amazon 리소스 이름)의 구성을 가져옵니다.

```
aws ivs-realtime get-composition \
    --arn "arn:aws:ivs:ap-northeast-1:123456789012:composition/wxyzWXYZpqrs"
```

출력:

```
{
    "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": {}
    }
}
```

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 복합 레코딩(실시간 스트리밍)을 참조하세요.

- 자세한 API 내용은 명령 참조GetComposition의 섹션을 참조하세요. AWS CLI

### get-encoder-configuration

다음 코드 예시에서는 get-encoder-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 인코더 구성을 가져오려면

다음 get-encoder-configuration 예제에서는 지정된 ARN (Amazon 리소스 이름)에서 지정한 구성 인코더 구성을 가져옵니다.

```
aws ivs-realtime get-encoder-configuration \
    --arn "arn:aws:ivs:ap-northeast-1:123456789012:encoder-configuration/
abcdABCDefgh"
```

출력:

```
{
    "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
        }
    }
}
```

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 Amazon IVS Stream에서 다중 호스트 활성화를 참조하세요.

- 자세한 API 내용은 명령 참조GetEncoderConfiguration의 섹션을 참조하세요. AWS CLI

### get-participant

다음 코드 예시에서는 get-participant을 사용하는 방법을 보여 줍니다.

AWS CLI

스테이지 참가자를 가져오려면

다음 get-participant 예제에서는 지정된 스테이지(ARNAmazon 리소스 이름)의 지정된 참가자 ID 및 세션 ID에 대한 스테이지 참가자를 가져옵니다.

```
aws ivs-realtime get-participant \
    --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
    --session-id st-a1b2c3d4e5f6g \
    --participant-id abCDEf12GHIj
```

출력:

```
{
    "participant": {
        "browserName", "Google Chrome",
        "browserVersion", "116",
        "firstJoinTime": "2023-04-26T20:30:34+00:00",
        "ispName", "Comcast",
        "osName", "Microsoft Windows 10 Pro",
        "osVersion", "10.0.19044"
        "participantId": "abCDEf12GHIj",
        "published": true,
        "recordingS3BucketName": "bucket-name",
        "recordingS3Prefix": "abcdABCDefgh/st-a1b2c3d4e5f6g/
abCDEf12GHIj/1234567890",
        "recordingState": "ACTIVE",
        "sdkVersion", "",
        "state": "CONNECTED",
        "userId": "",
    }
}
```

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 Amazon IVS Stream에서 다중 호스트 활성화를 참조하세요.

- 자세한 API 내용은 명령 참조GetParticipant의 섹션을 참조하세요. AWS CLI

### get-public-key

다음 코드 예시에서는 get-public-key을 사용하는 방법을 보여 줍니다.

AWS CLI

스테이지 참가자 토큰에 서명하는 데 사용되는 기존 퍼블릭 키를 가져오려면

다음 get-public-key 예제에서는 제공된 ARN에서 지정된 퍼블릭 키를 가져와서 스테이지 참가자 토큰을 사이징합니다.

```
aws ivs-realtime get-public-key \
    --arn arn:aws:ivs:us-west-2:123456789012:public-key/abcdABC1efg2
```

출력:

```
{
    "publicKey": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:public-key/abcdABC1efg2",
        "name": "",
        "publicKeyMaterial": "-----BEGIN PUBLIC KEY-----
\nMHYwEAYHKoZIzj0CAQYFK4EEACIDYgAEqVWUtqs6EktQMR1sCYmEzGvRwtaycI16\n9pmzcpiWu/
uhNStGlteJ5odRfRwVkoQUMnSZXTCcbn9bBTTmiWo4mJcFOOAzsthH
\n0UAb8NdD4tUE0At4a9hYP9IETEXAMPLE\n-----END PUBLIC KEY-----",
        "fingerprint": "12:a3:44:56:bc:7d:e8:9f:10:2g:34:hi:56:78:90:12",
        "tags": {}
    }
}
```

자세한 내용은 Amazon 실시간 스트리밍 사용 설명서의 참가자 토큰 배포를 참조하세요. IVS

- 자세한 API 내용은 명령 참조GetPublicKey의 섹션을 참조하세요. AWS CLI

### get-stage-session

다음 코드 예시에서는 get-stage-session을 사용하는 방법을 보여 줍니다.

AWS CLI

스테이지 세션을 가져오려면

다음 `get-stage-session` 예제에서는 지정된 단계의 지정된 세션 IDARN(Amazon 리소스 이름)에 대한 스테이지 세션을 가져옵니다.

```
aws ivs-realtime get-stage-session \
    --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
    --session-id st-a1b2c3d4e5f6g
```

출력:

```
{
    "stageSession": {
        "endTime": "2023-04-26T20:36:29+00:00",
        "sessionId": "st-a1b2c3d4e5f6g",
        "startTime": "2023-04-26T20:30:29.602000+00:00"
    }
}
```

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 Amazon IVS Stream에서 다중 호스트 활성화를 참조하세요.

- 자세한 API 내용은 명령 참조GetStageSession의 섹션을 참조하세요. AWS CLI

### `get-stage`

다음 코드 예시에서는 `get-stage`을 사용하는 방법을 보여 줍니다.

AWS CLI

스테이지의 구성 정보를 가져오려면

다음 `get-stage` 예제에서는 지정된 스테이지ARN(Amazon 리소스 이름)에 대한 스테이지 구성을 가져옵니다.

```
aws ivs-realtime get-stage \
    --arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh
```

출력:

```
{
    "stage": {
        "activeSessionId": "st-a1b2c3d4e5f6g",
        "arn": "arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh",
        "autoParticipantRecordingConfiguration": {
             "mediaTypes": [
                   "AUDIO_VIDEO"
             ],
             "storageConfigurationArn": "arn:aws:ivs:us-west-2:123456789012:storage-
configuration/abcdABCDefgh",
        },
        "endpoints": {
            "events": "wss://global.events.live-video.net",
            "whip": "https://1a2b3c4d5e6f.global-bm.whip.live-video.net"
        },
        "name": "test",
        "tags": {}
    }
}
```

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 Amazon IVS Stream에서 다중 호스트 활성화를 참조하세요.

- 자세한 API 내용은 명령 참조GetStage의 섹션을 참조하세요. AWS CLI

### `get-storage-configuration`

다음 코드 예시에서는 `get-storage-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 스토리지 구성을 가져오려면

다음 `get-storage-configuration` 예제에서는 지정된 ARN (Amazon 리소스 이름)에서 지정한 구성 스토리지 구성을 가져옵니다.

```
aws ivs-realtime get-storage-configuration \
    --name arn "arn:aws:ivs:ap-northeast-1:123456789012:storage-configuration/
abcdABCDefgh"
```

출력:

```
{
    "storageConfiguration": {
        "arn": "arn:aws:ivs:ap-northeast-1:123456789012:storage-configuration/
abcdABCDefgh",
        "name": "test-sc",
        "s3": {
            "bucketName": "test-bucket-name"
        },
        "tags": {}
    }
}
```

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 Amazon IVS Stream에서 다중 호스트 활성화를 참조하세요.

- 자세한 API 내용은 명령 참조GetStorageConfiguration의 섹션을 참조하세요. AWS CLI

### `import-public-key`

다음 코드 예시에서는 `import-public-key`을 사용하는 방법을 보여 줍니다.

AWS CLI

스테이지 참가자 토큰에 서명하는 데 사용할 기존 퍼블릭 키를 가져오려면

다음 `import-public-key` 예제에서는 스테이지 참가자 토큰을 크기 조정하는 데 사용할 퍼블릭 키를 자재 파일에서 가져옵니다.

```
aws ivs-realtime import-public-key \
    --public-key-material="`cat public.pem`"
```

출력:

```
{
    "publicKey": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:public-key/abcdABC1efg2",
        "name": "",
        "publicKeyMaterial": "-----BEGIN PUBLIC KEY-----
\nMHYwEAYHKoZIzj0CAQYFK4EEACIDYgAEqVWUtqs6EktQMR1sCYmEzGvRwtaycI16\n9pmzcpiWu/
uhNStGlteJ5odRfRwVkoQUMnSZXTCcbn9bBTTmiWo4mJcFOOAzsthH
\n0UAb8NdD4tUE0At4a9hYP9IETEXAMPLE\n-----END PUBLIC KEY-----",
        "fingerprint": "12:a3:44:56:bc:7d:e8:9f:10:2g:34:hi:56:78:90:12",
```

```
        "tags": {}
    }
}
```

자세한 내용은 Amazon 실시간 스트리밍 사용 설명서의 참가자 토큰 배포를 참조하세요. IVS

- 자세한 API 내용은 명령 참조ImportPublicKey의 섹션을 참조하세요. AWS CLI

### `list-compositions`

다음 코드 예시에서는 `list-compositions`을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 목록을 가져오려면

다음은 API 요청이 처리되는 AWS 리전의 AWS 계정에 대한 모든 구성을 `list-compositions` 나열합니다.

```
aws ivs-realtime list-compositions
```

출력:

```
{
    "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": {}
        }
    ]
}
```

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 Amazon IVS Stream에서 다중 호스트 활성화를 참조하세요.

- 자세한 API 내용은 명령 참조ListCompositions의 섹션을 참조하세요. AWS CLI

### `list-encoder-configurations`

다음 코드 예시에서는 `list-encoder-configurations`을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 인코더 구성을 나열하려면

다음은 API 요청이 처리되는 AWS 리전의 AWS 계정에 대한 모든 구성 인코더 구성을 `list-encoder-configurations` 나열합니다.

```
aws ivs-realtime list-encoder-configurations
```

출력:

```
{
    "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": {}
        }
    ]
}
```

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 Amazon IVS Stream에서 다중 호스트 활성화를 참조하세요.

- 자세한 API 내용은 명령 참조ListEncoderConfigurations의 섹션을 참조하세요. AWS CLI

### `list-participant-events`

다음 코드 예시에서는 `list-participant-events`을 사용하는 방법을 보여 줍니다.

AWS CLI

스테이지 참가자 이벤트 목록을 가져오려면

다음 `list-participant-events` 예제에서는 지정된 참가자 ID 및 지정된 단계의 세션 IDARN(Amazon 리소스 이름)에 대한 모든 참가자 이벤트를 나열합니다.

```
aws ivs-realtime list-participant-events \
    --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
    --session-id st-a1b2c3d4e5f6g \
    --participant-id abCDEf12GHIj
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 Amazon IVS Stream에서 다중 호스트 활성화를 참조하세요.

- 자세한 API 내용은 명령 참조ListParticipantEvents의 섹션을 참조하세요. AWS CLI

### `list-participants`

다음 코드 예시에서는 `list-participants`을 사용하는 방법을 보여 줍니다.

AWS CLI

스테이지 참가자 목록을 가져오려면

다음 `list-participants` 예제에서는 지정된 단계의 지정된 세션 IDARN(Amazon 리소스 이름)에 대한 모든 참가자를 나열합니다.

```
aws ivs-realtime list-participants \
    --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
```

```
    --session-id st-a1b2c3d4e5f6g
```

출력:

```
{
    "participants": [
        {
            "firstJoinTime": "2023-04-26T20:30:34+00:00",
            "participantId": "abCDEf12GHIj"
            "published": true,
            "recordingState": "STOPPED",
            "state": "DISCONNECTED",
            "userId": ""
        }
    ]
}
```

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 Amazon IVS Stream에서 다중 호스트 활성화를 참조하세요.

- 자세한 API 내용은 명령 참조ListParticipants의 섹션을 참조하세요. AWS CLI

### list-public-keys

다음 코드 예시에서는 list-public-keys을 사용하는 방법을 보여 줍니다.

AWS CLI

스테이지 참가자 토큰에 서명할 수 있는 기존 퍼블릭 키를 나열하려면

다음 list-public-keys 예제에서는 API 요청이 처리되는 AWS 리전의 스테이지 참가자 토큰을 차징하는 데 사용할 수 있는 모든 퍼블릭 키를 나열합니다.

```
aws ivs-realtime list-public-keys
```

출력:

```
{
    "publicKeys": [
        {
            "arn": "arn:aws:ivs:us-west-2:123456789012:public-key/abcdABC1efg2",
            "name": "",
```

```
            "tags": {}
        },
        {
            "arn": "arn:aws:ivs:us-west-2:123456789012:public-key/3bcdABCDefg4",
            "name": "",
            "tags": {}
        }
    ]
}
```

자세한 내용은 Amazon 실시간 스트리밍 사용 설명서의 참가자 토큰 배포를 참조하세요. IVS

- 자세한 API 내용은 명령 참조ListPublicKeys의 섹션을 참조하세요. AWS CLI

### `list-stage-sessions`

다음 코드 예시에서는 `list-stage-sessions`을 사용하는 방법을 보여 줍니다.

AWS CLI

스테이지 세션 목록을 가져오려면

다음 `list-stage-sessions` 예제에서는 지정된 단계ARN(Amazon 리소스 이름)의 모든 세션을 나열합니다.

```
aws ivs-realtime list-stage-sessions \
    --stage-arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh
```

출력:

```
{
    "stageSessions": [
        {
            "endTime": "2023-04-26T20:36:29+00:00",
            "sessionId": "st-a1b2c3d4e5f6g",
            "startTime": "2023-04-26T20:30:29.602000+00:00"
        }
    ]
}
```

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 Amazon IVS Stream에서 다중 호스트 활성화를 참조하세요.

- 자세한 API 내용은 명령 참조ListStageSessions의 섹션을 참조하세요. AWS CLI

### `list-stages`

다음 코드 예시에서는 `list-stages`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 단계에 대한 요약 정보를 가져오려면

다음 `list-stages` 예제에서는 API 요청이 처리되는 AWS 리전의 AWS 계정의 모든 단계를 나열합니다.

```
aws ivs-realtime list-stages
```

출력:

```
{
    "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": {}
        }
    ]
}
```

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 Amazon IVS Stream에서 다중 호스트 활성화를 참조하세요.

- 자세한 API 내용은 명령 참조ListStages의 섹션을 참조하세요. AWS CLI

### `list-storage-configurations`

다음 코드 예시에서는 `list-storage-configurations`을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 스토리지 구성을 나열하려면

다음은 API 요청이 처리되는 AWS 리전의 AWS 계정에 대한 모든 구성 스토리지 구성을 `list-storage-configurations` 나열합니다.

```
aws ivs-realtime list-storage-configurations
```

출력:

```
{
    "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": {}
        }
    ]
}
```

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 Amazon IVS Stream에서 다중 호스트 활성화를 참조하세요.

- 자세한 API 내용은 명령 참조ListStorageConfigurations의 섹션을 참조하세요. AWS CLI

### start-composition

다음 코드 예시에서는 start-composition을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 기본 레이아웃 설정으로 구성을 시작하려면

다음 start-composition 예제에서는 지정된 스테이지의 구성이 지정된 위치로 스트리밍되기 시작합니다.

```
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"}}]'
```

출력:

```
{
    "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": {}
    }
}
```

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 복합 레코딩(실시간 스트리밍)을 참조하세요.

예제 2: PiP 레이아웃으로 구성을 시작하려면

다음 `start-composition` 예제에서는 PiP 레이아웃을 사용하여 지정된 위치로 스트리밍되는 지정된 스테이지에 대한 구성을 시작합니다.

```
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"}'
```

출력:

```
{
    "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": {}
    }
}
```

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 복합 레코딩(실시간 스트리밍)을 참조하세요.

- 자세한 API 내용은 명령 참조StartComposition의 섹션을 참조하세요. AWS CLI

### stop-composition

다음 코드 예시에서는 stop-composition을 사용하는 방법을 보여 줍니다.

AWS CLI

구성을 중지하려면

다음은 지정된 ARN (Amazon 리소스 이름)에서 지정한 구성을 stop-composition 중지합니다.

```
aws ivs-realtime stop-composition \
    --arn "arn:aws:ivs:ap-northeast-1:123456789012:composition/abcdABCDefgh"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 Amazon IVS Stream에서 다중 호스트 활성화를 참조하세요.

- 자세한 API 내용은 명령 참조StopComposition의 섹션을 참조하세요. AWS CLI

### update-stage

다음 코드 예시에서는 update-stage을 사용하는 방법을 보여 줍니다.

AWS CLI

스테이지의 구성을 업데이트하려면

다음 update-stage 예제에서는 지정된 단계의 단계를 업데이트ARN하여 단계 이름을 업데이트하고 개별 참가자 레코딩을 구성합니다.

```
aws ivs-realtime update-stage \
    --arn arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh \
    --auto-participant-recording-configuration '{"mediaTypes":
 ["AUDIO_VIDEO"],"storageConfigurationArn": "arn:aws:ivs:us-
west-2:123456789012:storage-configuration/abcdABCDefgh"}' \
    --name stage1a
```

출력:

```
{
    "stage": {
        "arn": "arn:aws:ivs:us-west-2:123456789012:stage/abcdABCDefgh",
```

```
        "autoParticipantRecordingConfiguration": {
             "mediaTypes": [
                   "AUDIO_VIDEO"
             ],
             "storageConfigurationArn": "arn:aws:ivs:us-west-2:123456789012:storage-
configuration/abcdABCDefgh",
        },
        "endpoints": {
            "events": "wss://global.events.live-video.net",
            "whip": "https://1a2b3c4d5e6f.global-bm.whip.live-video.net"
        },
        "name": "stage1a",
        "tags": {}
    }
}
```

자세한 내용은 Amazon Interactive Video Service 사용 설명서의 Amazon IVS Stream에서 다중 호스트 활성화를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateStage의 섹션을 참조하세요. AWS CLI

## 를 사용한 Amazon Kendra 예제 AWS CLI

다음 코드 예제에서는 Amazon Kendra AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `create-data-source`

다음 코드 예시에서는 `create-data-source`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon Kendra 데이터 소스 커넥터를 생성하려면

다음은 Amazon Kendra 데이터 소스 커넥터를 `create-data-source` 생성하고 구성합니다. `describe-data-source` 를 사용하여 데이터 소스 커넥터의 상태를 확인하고, 상태가 데이터 소스 커넥터 “FAILED”를 표시하여 완전히 생성하는 경우 오류 메시지를 읽을 수 있습니다.

```
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"]}'
```

출력:

```
{
    "Id": "exampledatasource1"
}
```

자세한 내용은 Amazon Kendra 개발자 안내서의 Amazon Kendra 인덱스 및 데이터 소스 커넥터 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조CreateDataSource의 섹션을 참조하세요. AWS CLI

### create-index

다음 코드 예시에서는 create-index을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon Kendra 인덱스를 생성하려면

다음은 Amazon Kendra 인덱스를 create-index 생성하고 구성합니다. describe-index 를 사용하여 인덱스의 상태를 보고, 상태가 완전히 생성할 인덱스 'FAILED'를 표시하는 경우 오류 메시지를 읽을 수 있습니다.

```
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"}}'
```

출력:

```
{
   "Id": index1
}
```

자세한 내용은 Amazon Kendra 개발자 안내서의 Amazon Kendra 인덱스 및 데이터 소스 커넥터 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조CreateIndex의 섹션을 참조하세요. AWS CLI

### describe-data-source

다음 코드 예시에서는 describe-data-source을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon Kendra 데이터 소스 커넥터에 대한 정보를 가져오려면

다음은 Amazon Kendra 데이터 소스 커넥터에 대한 정보를 describe-data-source 가져옵니다. 데이터 소스 커넥터의 구성을 보고, 상태가 데이터 소스 커넥터 "FAILED"를 표시하여 완전히 생성하면 오류 메시지를 읽을 수 있습니다.

```
aws kendra describe-data-source \
    --id exampledatasource1 \
    --index-id exampleindex1
```

출력:

```
{
    "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"]
    }
}
```

자세한 내용은 Amazon Kendra 개발자 안내서의 Amazon Kendra 인덱스 및 데이터 소스 커넥터 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeDataSource의 섹션을 참조하세요. AWS CLI

### describe-index

다음 코드 예시에서는 describe-index을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon Kendra 인덱스에 대한 정보를 가져오려면

다음은 Amazon Kendra 인덱스에 대한 정보를 describe-index 가져옵니다. 인덱스의 구성을 보고, 상태가 완전히 생성하기 위해 인덱스 'FAILED'를 표시하는 경우 오류 메시지를 읽을 수 있습니다.

```
aws kendra describe-index \
    --id exampleindex1
```

출력:

```
{
    "CapacityUnits": {
        "QueryCapacityUnits": 0,
        "StorageCapacityUnits": 0
    },
    "CreatedAt": 2024-02-25T12:30:10+00:00,
    "Description": "Example index 1 contains the first set of example documents",
    "DocumentMetadataConfigurations": [
        {
            "Name": "_document_title",
            "Relevance": {
                "Importance": 8
            },
            "Search": {
                "Displayable": true,
                "Facetable": false,
                "Searchable": true,
                "Sortable": false
            },
            "Type": "STRING_VALUE"
        },
```

```
        {
            "Name": "_document_body",
            "Relevance": {
                "Importance": 5
            },
            "Search": {
                "Displayable": true,
                "Facetable": false,
                "Searchable": true,
                "Sortable": false
            },
            "Type": "STRING_VALUE"
        },
        {
            "Name": "_last_updated_at",
            "Relevance": {
                "Importance": 6,
                "Duration": "2628000s",
                "Freshness": true
            },
            "Search": {
                "Displayable": true,
                "Facetable": false,
                "Searchable": true,
                "Sortable": true
            },
            "Type": "DATE_VALUE"
        },
        {
            "Name": "department_custom_field",
            "Relevance": {
                "Importance": 7,
                "ValueImportanceMap": {
                    "Human Resources" : 4,
                    "Marketing and Sales" : 2,
                    "Research and innvoation" : 3,
                    "Admin" : 1
                }
            },
            "Search": {
                "Displayable": true,
                "Facetable": true,
                "Searchable": true,
                "Sortable": true
```

```
            },
            "Type": "STRING_VALUE"
        }
    ],
    "Edition": "DEVELOPER_EDITION",
    "Id": "index1",
    "IndexStatistics": {
        "FaqStatistics": {
            "IndexedQuestionAnswersCount": 10
        },
        "TextDocumentStatistics": {
            "IndexedTextBytes": 1073741824,
            "IndexedTextDocumentsCount": 1200
        }
    },
    "Name": "example index 1",
    "RoleArn": "arn:aws:iam::my-account-id:role/KendraRoleForExampleIndex",
    "ServerSideEncryptionConfiguration": {
        "KmsKeyId": "my-kms-key-id"
    },
    "Status": "ACTIVE",
    "UpdatedAt": 1709163615,
    "UserContextPolicy": "USER_TOKEN",
    "UserTokenConfigurations": [
        {
            "JsonTokenTypeConfiguration": {
                "GroupAttributeField": "groupNameField",
                "UserNameAttributeField": "userNameField"
            }
        }
    ]
}
```

자세한 내용은 Amazon Kendra 개발자 안내서의 Amazon Kendra 인덱스 및 데이터 소스 커넥터 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeIndex의 섹션을 참조하세요. AWS CLI

### update-data-source

다음 코드 예시에서는 update-data-source을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon Kendra 데이터 소스 커넥터를 업데이트하려면

다음은 Amazon Kendra 데이터 소스 커넥터의 구성을 update-data-source 업데이트합니다. 작업이 성공하면 서비스가 출력 없음, HTTP 상태 코드 200 또는 반환 코드 0을 AWS CLI 다시 보냅니다. describe-data-source 를 사용하여 데이터 소스 커넥터의 구성 및 상태를 볼 수 있습니다.

```
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"]}'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Kendra 개발자 안내서의 Amazon Kendra 인덱스 및 데이터 소스 커넥터 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateDataSource의 섹션을 참조하세요. AWS CLI

### update-index

다음 코드 예시에서는 update-index을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon Kendra 인덱스를 업데이트하려면

다음은 Amazon Kendra 인덱스의 구성을 update-index 업데이트합니다. 작업이 성공하면 서비스가 출력 없음, HTTP 상태 코드 200 또는 반환 코드 0을 AWS CLI 다시 보냅니다. describe-index 를 사용하여 인덱스의 구성 및 상태를 볼 수 있습니다.

```
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"}}'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Kendra 개발자 안내서의 Amazon Kendra 인덱스 및 데이터 소스 커넥터 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateIndex의 섹션을 참조하세요. AWS CLI

## 를 사용한 Kinesis 예제 AWS CLI

다음 코드 예제에서는 Kinesis와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `add-tags-to-stream`

다음 코드 예시에서는 `add-tags-to-stream`을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 스트림에 태그를 추가하려면

다음 add-tags-to-stream 예제에서는 키samplekey와 값이 있는 태그를 지정된 스트림example에 할당합니다.

```
aws kinesis add-tags-to-stream \
    --stream-name samplestream \
    --tags samplekey=example
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Kinesis Data Streams 개발자 안내서의 스트림 태그 지정을 참조하세요. Amazon Kinesis

- 자세한 API 내용은 명령 참조AddTagsToStream의 섹션을 참조하세요. AWS CLI

### create-stream

다음 코드 예시에서는 create-stream을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 스트림을 생성하는 방법

다음 create-stream 예시에서는 샤드 3개가 포함된 samplestream이라는 데이터 스트림을 생성합니다.

```
aws kinesis create-stream \
    --stream-name samplestream \
    --shard-count 3
```

이 명령은 출력을 생성하지 않습니다.

자세한 설명은 Amazon Kinesis Data Streams 개발자 안내서의 스트림 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateStream의 섹션을 참조하세요. AWS CLI

### decrease-stream-retention-period

다음 코드 예시에서는 decrease-stream-retention-period을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 스트림 보존 기간을 줄이려면

다음 `decrease-stream-retention-period` 예제에서는 samplestream이라는 스트림의 보존 기간(데이터 레코드가 스트림에 추가된 후 액세스할 수 있는 시간)을 48시간으로 줄입니다.

```
aws kinesis decrease-stream-retention-period \
    --stream-name samplestream \
    --retention-period-hours 48
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Kinesis Data Streams 개발자 안내서의 데이터 보존 기간 변경을 참조하세요.

- 자세한 API 내용은 명령 참조DecreaseStreamRetentionPeriod의 섹션을 참조하세요. AWS CLI

### `delete-stream`

다음 코드 예시에서는 `delete-stream`을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 스트림을 삭제하는 방법

다음 `delete-stream` 예시에서는 지정된 데이터 스트림을 삭제합니다.

```
aws kinesis delete-stream \
    --stream-name samplestream
```

이 명령은 출력을 생성하지 않습니다.

자세한 설명은 Amazon Kinesis Data Streams 개발자 안내서의 스트림 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteStream의 섹션을 참조하세요. AWS CLI

### `deregister-stream-consumer`

다음 코드 예시에서는 `deregister-stream-consumer`을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 스트림 소비자 등록을 취소하려면

다음 `deregister-stream-consumer` 예제에서는 지정된 데이터 스트림에서 지정된 소비자의 등록을 취소합니다.

```
aws kinesis deregister-stream-consumer \
    --stream-arn arn:aws:kinesis:us-west-2:123456789012:stream/samplestream \
    --consumer-name KinesisConsumerApplication
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Kinesis Data Streams 개발자 안내서의 Kinesis Data Streams를 사용하여 향상된 팬아웃을 사용하는 소비자 개발을 API 참조하세요. Amazon Kinesis

- 자세한 API 내용은 명령 참조DeregisterStreamConsumer의 섹션을 참조하세요. AWS CLI

### `describe-limits`

다음 코드 예시에서는 `describe-limits`을 사용하는 방법을 보여 줍니다.

AWS CLI

샤드 제한을 설명하려면

다음 `describe-limits` 예제에서는 현재 AWS 계정의 샤드 제한 및 사용량을 보여줍니다.

```
aws kinesis describe-limits
```

출력:

```
{
    "ShardLimit": 500,
    "OpenShardCount": 29
}
```

자세한 내용은 Amazon Kinesis Data Streams 개발자 안내서의 스트림 공유를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeLimits의 섹션을 참조하세요. AWS CLI

### describe-stream-consumer

다음 코드 예시에서는 describe-stream-consumer을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 스트림 소비자를 설명하려면

다음 describe-stream-consumer 예제에서는 지정된 데이터 스트림에 등록된 지정된 소비자에 대한 설명을 반환합니다.

```
aws kinesis describe-stream-consumer \
    --stream-arn arn:aws:kinesis:us-west-2:012345678912:stream/samplestream \
    --consumer-name KinesisConsumerApplication
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Kinesis Data Streams 개발자 안내서의 Amazon Kinesis Data Streams에서 데이터 읽기를 참조하세요. Amazon Kinesis

- 자세한 API 내용은 명령 참조DescribeStreamConsumer의 섹션을 참조하세요. AWS CLI

### describe-stream-summary

다음 코드 예시에서는 describe-stream-summary을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 스트림 요약을 설명하려면

다음 describe-stream-summary 예제에서는 지정된 데이터 스트림에 대한 요약된 설명(샤드 목록 제외)을 제공합니다.

```
aws kinesis describe-stream-summary \
    --stream-name samplestream
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 Amazon Kinesis Data Streams 개발자 안내서의 스트림 생성 및 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeStreamSummary의 섹션을 참조하세요. AWS CLI

### describe-stream

다음 코드 예시에서는 describe-stream을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 스트림을 설명하는 방법

다음 describe-stream 예시에서는 지정된 데이터 스트림의 세부 정보를 반환합니다.

```
aws kinesis describe-stream \
    --stream-name samplestream
```

출력:

```
{
    "StreamDescription": {
        "Shards": [
            {
                "ShardId": "shardId-000000000000",
                "HashKeyRange": {
                    "StartingHashKey": "0",
                    "EndingHashKey": "113427455640312821154458202477256070484"
                },
                "SequenceNumberRange": {
                    "StartingSequenceNumber":
 "496008716829570364423650249261910734372510605801286533314"
                }
            },
            {
                "ShardId": "shardId-000000000001",
                "HashKeyRange": {
                    "StartingHashKey": "113427455640312821154458202477256070485",
                    "EndingHashKey": "226854911280625642308916404954512140969"
                },
                "SequenceNumberRange": {
                    "StartingSequenceNumber":
 "496008716829793371875635555493326091555237089416346337746"
                }
            },
            {
                "ShardId": "shardId-000000000002",
                "HashKeyRange": {
                    "StartingHashKey": "226854911280625642308916404954512140970",
                    "EndingHashKey": "340282366920938463463374607431768211455"
                },
                "SequenceNumberRange": {
                    "StartingSequenceNumber":
 "496008716830016379327620861724741448737963573031406141 78"
                }
            }
        ],
        "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
    }
}
```

자세한 내용은 Amazon Kinesis Data Streams 개발자 안내서의 스트림 생성 및 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeStream의 섹션을 참조하세요. AWS CLI

### disable-enhanced-monitoring

다음 코드 예시에서는 disable-enhanced-monitoring을 사용하는 방법을 보여 줍니다.

AWS CLI

샤드 수준 지표에 대한 향상된 모니터링을 비활성화하려면

다음 disable-enhanced-monitoring 예제에서는 샤드 수준 지표에 대한 향상된 Kinesis 데이터 스트림 모니터링을 비활성화합니다.

```
aws kinesis disable-enhanced-monitoring \
    --stream-name samplestream --shard-level-metrics ALL
```

출력:

```
{
    "StreamName": "samplestream",
    "CurrentShardLevelMetrics": [
        "IncomingBytes",
        "OutgoingRecords",
        "IteratorAgeMilliseconds",
        "IncomingRecords",
        "ReadProvisionedThroughputExceeded",
        "WriteProvisionedThroughputExceeded",
        "OutgoingBytes"
    ],
    "DesiredShardLevelMetrics": []
}
```

자세한 내용은 Amazon Kinesis Data Streams 개발자 안내서의 Amazon Kinesis Data Streams에서 스트림 모니터링을 참조하세요. Amazon Kinesis

- 자세한 API 내용은 명령 참조DisableEnhancedMonitoring의 섹션을 참조하세요. AWS CLI

### `enable-enhanced-monitoring`

다음 코드 예시에서는 `enable-enhanced-monitoring`을 사용하는 방법을 보여 줍니다.

AWS CLI

샤드 수준 지표에 대한 향상된 모니터링을 활성화하려면

다음 `enable-enhanced-monitoring` 예제에서는 샤드 수준 지표에 대한 향상된 Kinesis 데이터 스트림 모니터링을 활성화합니다.

```
aws kinesis enable-enhanced-monitoring \
    --stream-name samplestream \
    --shard-level-metrics ALL
```

출력:

```
{
    "StreamName": "samplestream",
    "CurrentShardLevelMetrics": [],
    "DesiredShardLevelMetrics": [
        "IncomingBytes",
        "OutgoingRecords",
        "IteratorAgeMilliseconds",
        "IncomingRecords",
        "ReadProvisionedThroughputExceeded",
        "WriteProvisionedThroughputExceeded",
        "OutgoingBytes"
    ]
}
```

자세한 내용은 Amazon Kinesis Data Streams 개발자 안내서의 Amazon Kinesis Data Streams에서 스트림 모니터링을 참조하세요. Amazon Kinesis

- 자세한 API 내용은 명령 참조EnableEnhancedMonitoring의 섹션을 참조하세요. AWS CLI

### `get-records`

다음 코드 예시에서는 `get-records`을 사용하는 방법을 보여 줍니다.

AWS CLI

샤드에서 레코드를 가져오는 방법

다음 `get-records` 예시에서는 지정된 샤드 이터레이터를 사용하여 Kinesis 데이터 스트림의 샤드에서 데이터 레코드를 가져옵니다.

```
aws kinesis get-records \
    --shard-iterator AAAAAAAAAAF7/0mWD7IuHj1yGv/
TKuNgx2ukD5xipCY4cy4gU96orWwZwcSXh3K9tAmGYeOZyLZrvzzeOFVf9iN99hUPw/w/
b0YWYeehfNvnf1DYt5XpDJghLKr3DzgznkTmMymDP3R+3wRKeuEw6/kdxY2yKJH0veaiekaVc4N2VwK/
GvaGP2Hh9Fg7N++q0Adg6fIDQPt4p8RpavDbk+A4sL9SWGE1
```

출력:

```
{
    "Records": [],
    "MillisBehindLatest": 80742000
}
```

자세한 내용은 Amazon Kinesis Data Streams 개발자 안내서의 Java용 API에서 AWS SDK Kinesis Data Streams를 사용하는 소비자 개발을 참조하세요. Amazon Kinesis

- 자세한 API 내용은 명령 참조GetRecords의 섹션을 참조하세요. AWS CLI

### `get-shard-iterator`

다음 코드 예시에서는 `get-shard-iterator`을 사용하는 방법을 보여 줍니다.

AWS CLI

샤드 반복자를 가져오려면

다음 `get-shard-iterator` 예제에서는 AT_SEQUENCE_NUMBER 샤드 반복자 유형을 사용하고 샤드 반복자를 생성하여 지정된 시퀀스 번호로 표시된 위치에서 데이터 레코드를 정확히 읽기 시작합니다.

```
aws kinesis get-shard-iterator \
    --stream-name samplestream \
    --shard-id shardId-000000000001 \
    --shard-iterator-type LATEST
```

출력:

```
{
    "ShardIterator": "AAAAAAAAAAFEvJjIYI+3jw/4aqgH9FifJ+n48XWTh/
IFIsbILP6o5eDueD39NXNBfpZ10WL5K6ADXk8w+5H+Qhd9cFA9k268CPXCz/kebq1TGYI7Vy
+lUkA9BuN3xvATxMBGxRY3zYK05gqgvaIRn94O8SqeEqwhigwZxNWxID3Ej7YYYcxQi8Q/fIrCjGAy/
n2r5Z9G864YpWDfN9upNNQAR/iiOWKs"
}
```

자세한 내용은 Amazon Kinesis Data Streams 개발자 안내서의 Java용 API에서 AWS SDK Kinesis Data Streams를 사용하여 소비자 개발을 참조하세요. Amazon Kinesis

- 자세한 API 내용은 명령 참조GetShardIterator의 섹션을 참조하세요. AWS CLI

### increase-stream-retention-period

다음 코드 예시에서는 increase-stream-retention-period을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 스트림 보존 기간을 늘리려면

다음 increase-stream-retention-period 예제에서는 지정된 스트림의 보존 기간(데이터 레코드가 스트림에 추가된 후 액세스할 수 있는 시간)을 168시간으로 늘립니다.

```
aws kinesis increase-stream-retention-period \
    --stream-name samplestream \
    --retention-period-hours 168
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Kinesis Data Streams 개발자 안내서의 데이터 보존 기간 변경을 참조하세요.

- 자세한 API 내용은 명령 참조IncreaseStreamRetentionPeriod의 섹션을 참조하세요. AWS CLI

### list-shards

다음 코드 예시에서는 list-shards을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 스트림의 샤드를 나열하려면

다음 list-shards 예제에서는 지정된 스트림의 모든 샤드를 나열합니다. 샤드는 ID가 지정된 exclusive-start-shard-id를 바로 따르는 샤드로 시작합니다shardId-000000000000.

```
aws kinesis list-shards \
    --stream-name samplestream \
    --exclusive-start-shard-id shardId-000000000000
```

출력:

```
{
    "Shards": [
        {
            "ShardId": "shardId-000000000001",
            "HashKeyRange": {
                "StartingHashKey": "113427455640312821154458202477256070485",
                "EndingHashKey": "226854911280625642308916404954512140969"
            },
            "SequenceNumberRange": {
                "StartingSequenceNumber":
 "496008716829793371875635555493326091555237089416346337 46"
            }
        },
        {
            "ShardId": "shardId-000000000002",
            "HashKeyRange": {
                "StartingHashKey": "226854911280625642308916404954512140970",
                "EndingHashKey": "340282366920938463463374607431768211455"
            },
            "SequenceNumberRange": {
                "StartingSequenceNumber":
 "496008716830016379327620861724741448737963573031406141 78"
            }
        }
    ]
}
```

자세한 내용은 Amazon Kinesis Data Streams 개발자 안내서의 샤드 목록을 참조하세요. Amazon Kinesis

- 자세한 API 내용은 명령 참조ListShards의 섹션을 참조하세요. AWS CLI

### `list-streams`

다음 코드 예시에서는 `list-streams`을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 스트림을 나열하는 방법

다음 `list-streams` 예시에서는 현재 계정 및 리전의 모든 활성 데이터 스트림을 나열합니다.

```
aws kinesis list-streams
```

출력:

```
{
    "StreamNames": [
        "samplestream",
        "samplestream1"
    ]
}
```

자세한 내용은 Amazon Kinesis Data Streams 개발자 안내서의 스트림 나열을 참조하세요.

- 자세한 API 내용은 명령 참조ListStreams의 섹션을 참조하세요. AWS CLI

### `list-tags-for-stream`

다음 코드 예시에서는 `list-tags-for-stream`을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 스트림의 태그를 나열하려면

다음 `list-tags-for-stream` 예제에서는 지정된 데이터 스트림에 연결된 태그를 나열합니다.

```
aws kinesis list-tags-for-stream \
```

```
    --stream-name samplestream
```

출력:

```
{
    "Tags": [
        {
            "Key": "samplekey",
            "Value": "example"
        }
    ],
    "HasMoreTags": false
}
```

자세한 내용은 Amazon Kinesis Data Streams 개발자 안내서의 스트림 태그 지정을 참조하세요. Amazon Kinesis

- 자세한 API 내용은 명령 참조ListTagsForStream의 섹션을 참조하세요. AWS CLI

### merge-shards

다음 코드 예시에서는 merge-shards을 사용하는 방법을 보여 줍니다.

AWS CLI

샤드를 병합하려면

다음 merge-shards 예제에서는 두 개의 인접한 샤드를 지정된 데이터 스트림IDs의 shardId-000000000000 및 shardId-000000000001과 병합하고 이를 단일 샤드로 결합합니다.

```
aws kinesis merge-shards \
    --stream-name samplestream \
    --shard-to-merge shardId-000000000000 \
    --adjacent-shard-to-merge shardId-000000000001
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Kinesis Data Streams 개발자 안내서의 두 개의 샤드 병합을 참조하세요. Amazon Kinesis

- 자세한 API 내용은 명령 참조MergeShards의 섹션을 참조하세요. AWS CLI

### put-record

다음 코드 예시에서는 put-record을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 스트림에 레코드를 쓰는 방법

다음 put-record 예시에서는 지정된 파티션 키를 사용하여 지정된 데이터 스트림에 단일 데이터 레코드를 씁니다.

```
aws kinesis put-record \
    --stream-name samplestream \
    --data sampledatarecord \
    --partition-key samplepartitionkey
```

출력:

```
{
    "ShardId": "shardId-000000000009",
    "SequenceNumber": "49600902273357540915989931256901506243878407835297513618",
    "EncryptionType": "KMS"
}
```

자세한 내용은 Amazon Kinesis Data Streams 개발자 안내서의 Java용 API에서 AWS SDK Amazon Kinesis Data Streams를 사용하여 생산자 개발을 참조하세요. Amazon Kinesis

- 자세한 API 내용은 명령 참조PutRecord의 섹션을 참조하세요. AWS CLI

### put-records

다음 코드 예시에서는 put-records을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 스트림에 여러 레코드를 쓰려면

다음 put-records 예제에서는 지정된 파티션 키를 사용하여 데이터 레코드를 작성하고 단일 호출에서 다른 파티션 키를 사용하여 다른 데이터 레코드를 작성합니다.

```
aws kinesis put-records \
```

```
    --stream-name samplestream \
    --
records Data=blob1,PartitionKey=partitionkey1 Data=blob2,PartitionKey=partitionkey2
```

출력:

```
{
    "FailedRecordCount": 0,
    "Records": [
        {
            "SequenceNumber":
 "49600883331171471519674795588238531498465399900093808706",
            "ShardId": "shardId-000000000004"
        },
        {
            "SequenceNumber":
 "49600902273357540915989931256902715169698037101720764562",
            "ShardId": "shardId-000000000009"
        }
    ],
    "EncryptionType": "KMS"
}
```

자세한 내용은 Amazon Kinesis Data Streams 개발자 안내서의 Java용 API에서 AWS SDK Amazon Kinesis Data Streams를 사용하여 생산자 개발을 참조하세요. Amazon Kinesis

- 자세한 API 내용은 명령 참조PutRecords의 섹션을 참조하세요. AWS CLI

### register-stream-consumer

다음 코드 예시에서는 register-stream-consumer을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 스트림 소비자를 등록하려면

다음 register-stream-consumer 예제에서는 에서 호출한 소비자를 지정된 데이터 스트림KinesisConsumerApplication에 등록합니다.

```
aws kinesis register-stream-consumer \
    --stream-arn arn:aws:kinesis:us-west-2:012345678912:stream/samplestream \
    --consumer-name KinesisConsumerApplication
```

출력:

```
{
    "Consumer": {
        "ConsumerName": "KinesisConsumerApplication",
        "ConsumerARN": "arn:aws:kinesis:us-west-2: 123456789012:stream/samplestream/
consumer/KinesisConsumerApplication:1572383852",
        "ConsumerStatus": "CREATING",
        "ConsumerCreationTimestamp": 1572383852.0
    }
}
```

자세한 내용은 Amazon Kinesis Data Streams 개발자 안내서의 Kinesis Data Streams를 사용하여 향상된 팬아웃을 사용하는 소비자 개발을 API 참조하세요. Amazon Kinesis

- 자세한 API 내용은 명령 참조RegisterStreamConsumer의 섹션을 참조하세요. AWS CLI

### `remove-tags-from-stream`

다음 코드 예시에서는 `remove-tags-from-stream`을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 스트림에서 태그를 제거하려면

다음 `remove-tags-from-stream` 예제에서는 지정된 데이터 스트림에서 지정된 키가 있는 태그를 제거합니다.

```
aws kinesis remove-tags-from-stream \
    --stream-name samplestream \
    --tag-keys samplekey
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Kinesis Data Streams 개발자 안내서의 스트림 태그 지정을 참조하세요. Amazon Kinesis

- 자세한 API 내용은 명령 참조RemoveTagsFromStream의 섹션을 참조하세요. AWS CLI

### `split-shard`

다음 코드 예시에서는 `split-shard`을 사용하는 방법을 보여 줍니다.

AWS CLI

샤드를 분할하려면

다음 split-shard 예제에서는 새 시작 해시 키 10을 사용하여 지정된 샤드를 두 개의 새 샤드로 분할합니다.

```
aws kinesis split-shard \
    --stream-name samplestream \
    --shard-to-split shardId-000000000000 \
    --new-starting-hash-key 10
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Kinesis Data Streams 개발자 안내서의 샤드 분할을 참조하세요.

- 자세한 API 내용은 명령 참조SplitShard의 섹션을 참조하세요. AWS CLI

### start-stream-encryption

다음 코드 예시에서는 start-stream-encryption을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 스트림 암호화를 활성화하려면

다음 start-stream-encryption 예제에서는 지정된 키를 사용하여 지정된 AWS KMS 스트림에 대한 서버 측 암호화를 활성화합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Kinesis Data Streams 개발자 안내서의 Amazon Kinesis Data Streams의 데이터 보호를 참조하세요. Amazon Kinesis

- 자세한 API 내용은 명령 참조StartStreamEncryption의 섹션을 참조하세요. AWS CLI

### stop-stream-encryption

다음 코드 예시에서는 stop-stream-encryption을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 스트림 암호화를 비활성화하려면

다음 stop-stream-encryption 예제에서는 지정된 키를 사용하여 지정된 AWS KMS 스트림에 대한 서버 측 암호화를 비활성화합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Kinesis Data Streams 개발자 안내서의 Amazon Kinesis Data Streams의 데이터 보호를 참조하세요. Amazon Kinesis

- 자세한 API 내용은 명령 참조StopStreamEncryption의 섹션을 참조하세요. AWS CLI

### update-shard-count

다음 코드 예시에서는 update-shard-count을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 스트림에서 샤드 수를 업데이트하려면

다음 update-shard-count 예제에서는 지정된 데이터 스트림의 샤드 수를 6으로 업데이트합니다. 이 예제에서는 동일한 크기의 샤드를 생성하는 균일한 크기 조정을 사용합니다.

```
aws kinesis update-shard-count \
    --stream-name samplestream \
    --scaling-type UNIFORM_SCALING \
    --target-shard-count 6
```

출력:

```
{
```

```
    "StreamName": "samplestream",
    "CurrentShardCount": 3,
    "TargetShardCount": 6
}
```

자세한 내용은 Amazon Kinesis Data Streams 개발자 안내서의 스트림 공유를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateShardCount의 섹션을 참조하세요. AWS CLI

# AWS KMS 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS KMS.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### `cancel-key-deletion`

다음 코드 예시에서는 `cancel-key-deletion`을 사용하는 방법을 보여 줍니다.

AWS CLI

고객 관리형 KMS 키의 예약된 삭제를 취소하려면

다음 `cancel-key-deletion` 예제에서는 고객 관리형 KMS 키의 예약된 삭제를 취소합니다.

```
aws kms cancel-key-deletion \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

출력:

```
{
    "KeyId": "arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab"
}
```

cancel-key-deletion 명령이 성공하면 예약된 삭제가 취소됩니다. 그러나 KMS 키의 키 상태는 Disabled이므로 암호화 작업에서 KMS 키를 사용할 수 없습니다. 기능을 복원하려면 enable-key 명령을 사용합니다.

자세한 내용은 Key Management Service 개발자 안내서의 키 삭제 예약 및 취소를 참조하세요. AWS

- 자세한 API 내용은 명령 참조CancelKeyDeletion의 섹션을 참조하세요. AWS CLI

### connect-custom-key-store

다음 코드 예시에서는 connect-custom-key-store을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 키 스토어를 연결하려면

다음 connect-custom-key-store 예제에서는 지정된 사용자 지정 키 스토어를 다시 연결합니다. 이 명령과 같은 명령을 사용하여 사용자 지정 키 스토어를 처음 연결하거나 연결이 끊긴 키 스토어를 다시 연결할 수 있습니다.

이 명령을 사용하여 AWS CloudHSM 키 스토어 또는 외부 키 스토어를 연결할 수 있습니다.

```
aws kms connect-custom-key-store \
    --custom-key-store-id cks-1234567890abcdef0
```

이 명령은 출력을 반환하지 않습니다. 명령이 유효한지 확인하려면 describe-custom-key-stores 명령을 사용하세요.

AWS CloudHSM 키 스토어 연결에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 AWS CloudHSM 키 스토어 연결 및 연결을 참조하세요.

외부 키 스토어 연결에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 외부 키 스토어 연결 및 연결 해제를 참조하세요.

- 자세한 API 내용은 명령 참조ConnectCustomKeyStore의 섹션을 참조하세요. AWS CLI

### create-alias

다음 코드 예시에서는 create-alias을 사용하는 방법을 보여 줍니다.

AWS CLI

KMS 키의 별칭을 생성하려면

다음 create-alias 명령은 KMS 키 ID 로 식별되는 키example-alias의 이름이 인 별칭을 생성합니다1234abcd-12ab-34cd-56ef-1234567890ab.

별칭 이름은 alias/로 시작해야 합니다. 로 시작하는 별칭 이름은 사용하지 마세요. alias/aws에서 사용하도록 예약되어 있습니다 AWS.

```
aws kms create-alias \
    --alias-name alias/example-alias \
    --target-key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

이 명령은 출력을 반환하지 않습니다. 새 별칭을 보려면 list-aliases 명령을 사용하세요.

자세한 내용은 AWS Key Management Service 개발자 안내서의 별칭 사용을 참조하세요.

- 자세한 API 내용은 명령 참조CreateAlias의 섹션을 참조하세요. AWS CLI

### create-custom-key-store

다음 코드 예시에서는 create-custom-key-store을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: AWS CloudHSM 키 스토어 생성

다음 create-custom-key-store 예제에서는 필요한 파라미터를 사용하여 AWS 클라우드HSM 클러스터가 지원하는 AWS 클라우드HSM 키 스토어를 생성합니다. 를 추가할 수도 있습니다custom-key-store-type``parameter with the default value: ``AWS_CLOUDHSM.

에서 trust-anchor-certificate 명령에 대한 파일 입력을 지정하려면 AWS CLI file:// 접두사가 필요합니다.

```
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
```

출력:

```
{
    "CustomKeyStoreId": cks-1234567890abcdef0
}
```

자세한 내용은 Key Management Service 개발자 안내서의 AWS 클라우드HSM 키 스토어 생성을 참조하세요. AWS

예제 2: 퍼블릭 엔드포인트 연결을 사용하여 외부 키 스토어를 생성하려면

다음 `create-custom-key-store` 예제에서는 인터넷을 통해 와 AWS KMS 통신하는 외부 키 스토어(XKS)를 생성합니다.

이 예제에서 는 의 선택적 접두사를 `XksProxyUriPath` 사용합니다`example-prefix`.

NOTE: 버전 1.0을 사용하는 AWS CLI 경우 파라미터와 같은 HTTP 또는 HTTPS 값으로 XksProxyUriEndpoint 파라미터를 지정하기 전에 다음 명령을 실행합니다.

```
aws configure set cli_follow_urlparam false
```

그렇지 않으면 AWS CLI 버전 1.0이 파라미터 값을 해당 URI 주소에 있는 콘텐츠로 바꿉니다.

```
aws kms create-custom-key-store \
    --custom-key-store-name ExamplePublicEndpointXKS \
    --custom-key-store-type EXTERNAL_KEY_STORE \
    --xks-proxy-connectivity PUBLIC_ENDPOINT \
    --xks-proxy-uri-endpoint "https://myproxy.xks.example.com" \
    --xks-proxy-uri-path "/example-prefix/kms/xks/v1" \
    --xks-proxy-authentication-credential "AccessKeyId=ABCDE12345670EXAMPLE,
 RawSecretAccessKey=DXjSUawnel2fr6SKC7G25CNxTyWKE5PF9XX6H/u9pSo="
```

출력:

```
{
    "CustomKeyStoreId": cks-2234567890abcdef0
}
```

자세한 내용은 Key Management Service 개발자 안내서의 외부 키 스토어 생성을 참조하세요. AWS

예제 3: VPC 엔드포인트 서비스 연결을 사용하여 외부 키 스토어를 생성하려면

다음 `create-custom-key-store` 예제에서는 Amazon VPC 엔드포인트 서비스를 사용하여 와 통신하는 외부 키 스토어(XKS)를 생성합니다 AWS KMS.

NOTE: 버전 1.0을 사용하는 AWS CLI 경우 파라미터와 같은 HTTP 또는 HTTPS 값으로 XksProxyUriEndpoint 파라미터를 지정하기 전에 다음 명령을 실행합니다.

```
aws configure set cli_follow_urlparam false
```

그렇지 않으면 AWS CLI 버전 1.0이 파라미터 값을 해당 URI 주소에 있는 콘텐츠로 바꿉니다.

```
aws kms create-custom-key-store \
    --custom-key-store-name ExampleVPCEndpointXKS \
    --custom-key-store-type EXTERNAL_KEY_STORE \
    --xks-proxy-connectivity VPC_ENDPOINT_SERVICE \
    --xks-proxy-uri-endpoint "https://myproxy-private.xks.example.com" \
    --xks-proxy-uri-path "/kms/xks/v1" \
    --xks-proxy-vpc-endpoint-service-name "com.amazonaws.vpce.us-east-1.vpce-svc-
example1" \
    --xks-proxy-authentication-credential "AccessKeyId=ABCDE12345670EXAMPLE,
 RawSecretAccessKey=DXjSUawnel2fr6SKC7G25CNxTyWKE5PF9XX6H/u9pSo="
```

출력:

```
{
    "CustomKeyStoreId": cks-3234567890abcdef0
}
```

자세한 내용은 Key Management Service 개발자 안내서의 외부 키 스토어 생성을 참조하세요. AWS

- 자세한 API 내용은 명령 참조CreateCustomKeyStore의 섹션을 참조하세요. AWS CLI

### `create-grant`

다음 코드 예시에서는 `create-grant`을 사용하는 방법을 보여 줍니다.

AWS CLI

권한 부여를 생성하는 방법

다음 create-grant 예제에서는 exampleUser 사용자가 1234abcd-12ab-34cd-56ef-1234567890ab 예제 KMS 키에서 decrypt 명령을 사용할 수 있는 권한을 생성합니다. 사용 중지하는 보안 주체는 adminRole 역할입니다. 이 권한 부여는 decrypt 요청의 암호화 컨텍스트에 "Department": "IT" 키-값 페어가 포함된 경우에만 이 권한을 허용하도록 EncryptionContextSubset 권한 부여 제약 조건을 사용합니다.

```
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
```

출력:

```
{
    "GrantId": "1a2b3c4d2f5e69f440bae30eaec9570bb1fb7358824f9ddfa1aa5a0dab1a59b2",
    "GrantToken": "<grant token here>"
}
```

권한 부여에 대한 자세한 정보를 보려면 list-grants 명령을 사용하세요.

자세한 내용은 AWS Key Management Service 개발자 안내서의 에서 권한 부여 AWS KMS를 참조하세요.

- 자세한 API 내용은 명령 참조CreateGrant의 섹션을 참조하세요. AWS CLI

### create-key

다음 코드 예시에서는 create-key을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 에서 고객 관리형 KMS 키를 생성하려면 AWS KMS

다음 create-key 예제에서는 대칭 암호화 KMS 키를 생성합니다.

대칭 암호화 KMS 키인 기본 키를 생성하려면 파라미터를 지정할 필요가 없습니다. 이러한 파라미터의 기본값은 대칭 암호화 키를 생성합니다.

이 명령은 키 정책을 지정하지 않으므로 프로그래밍 방식으로 생성된 KMS 키에 대한 기본 키 정책을 키에 가져옵니다. KMS 키 정책을 보려면 `get-key-policy` 명령을 사용하세요. 키 정책을 변경하려면 `put-key-policy` 명령을 사용하세요.

```
aws kms create-key
```

`create-key` 명령은 키 ID 및 새 KMS 키ARN의 키 메타데이터를 반환합니다. 이러한 값을 사용하여 다른 AWS KMS 작업의 KMS 키를 식별할 수 있습니다. 출력에 태그가 포함되지 않습니다. KMS 키의 태그를 보려면 를 사용합니다`list-resource-tags command`.

출력:

```
{
    "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"
        ]
    }
}
```

참고: 이 `create-key` 명령을 사용하면 별칭을 지정할 수 없습니다. 새 KMS 키에 대한 별칭을 생성하려면 `create-alias` 명령을 사용합니다.

자세한 내용은 AWS Key Management Service 개발자 안내서의 키 생성을 참조하세요.

예제 2: 암호화 및 복호화를 위한 비대칭 RSA KMS 키 생성

다음 create-key 예제에서는 암호화 및 복호화를 위한 비대칭 키 페어가 KMS 포함된 RSA 키를 생성합니다.

```
aws kms create-key \
   --key-spec RSA_4096 \
   --key-usage ENCRYPT_DECRYPT
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Key AWS Management Service 개발자 안내서의 에서 비대칭 AWS KMS 키를 참조하세요.

예제 3: 서명 및 확인을 위한 비대칭 타원 곡선 KMS 키 생성

서명 및 확인을 위한 비대칭 타원 곡선(ECC) KMS 키 페어가 포함된 비대칭 키를 생성합니다. 가 ECC KMS 키에 유효한 유일한 값인 경우에도 --key-usage 파라미터SIGN_VERIFY가 필요합니다.

```
aws kms create-key \
    --key-spec ECC_NIST_P521 \
    --key-usage SIGN_VERIFY
```

출력:

```
{
    "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"
        ]
    }
}
```

자세한 내용은 Key AWS Management Service 개발자 안내서의 에서 비대칭 AWS KMS 키를 참조하세요.

예제 4: HMAC KMS 키를 생성하려면

다음 `create-key` 예제에서는 384비트 HMAC KMS 키를 생성합니다. `--key-usage` 파라미터의 `GENERATE_VERIFY_MAC` 값은 HMAC KMS 키에 대해 유효한 유일한 값이지만 필수입니다.

```
aws kms create-key \
    --key-spec HMAC_384 \
    --key-usage GENERATE_VERIFY_MAC
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 HMAC Key Management Service 개발자 안내서의 AWS KMSAWS 에서 키를 참조하세요.

예제 4: 다중 리전 기본 KMS 키를 생성하려면

다음 create-key 예시에서는 다중 리전 프라이머리 대칭 암호화 키를 생성합니다. 모든 파라미터의 기본값은 대칭 암호화 키를 생성하기 때문에 이 KMS 키에는 --multi-region 파라미터만 필요합니다. 에서 부울 파라미터가 true임을 나타내 AWS CLI려면 파라미터 이름을 지정하면 됩니다.

```
aws kms create-key \
    --multi-region
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS Key Management Service 개발자 안내서의 에서 비대칭 키를 AWS KMS 참조하세요.

예제 5: 가져온 KMS 키 구성 요소에 대한 키를 생성하려면

다음 `create-key` 예제에서는 KMS 키 구성 요소가 없는 키를 생성합니다. 작업이 완료되면 자체 키 구성 요소를 KMS 키로 가져올 수 있습니다. 이 KMS 키를 생성하려면 `--origin` 파라미터를 로 설정합니다EXTERNAL.

```
aws kms create-key \
    --origin EXTERNAL
```

출력:

```
{
     "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"
    }
}
```

자세한 내용은 Key Management Service 개발자 안내서의 키에서 AWS KMS 키 구성 요소 가져오기를 참조하세요. AWS

예제 6: AWS Cloud KMS HSM 키 스토어에서 키를 생성하려면

다음 `create-key` 예제에서는 지정된 AWS Cloud KMS HSM 키 스토어에서 가 키를 생성합니다. 작업은 에서 KMS AWS KMS 키와 메타데이터를 생성하고 사용자 지정 키 스토어와 연결된 AWS 클라우드HSM 클러스터에 키 구성 요소를 생성합니다. `--custom-key-store-id` 및 `--origin` 파라미터가 필요합니다.

```
aws kms create-key \
    --origin AWS_CLOUDHSM \
    --custom-key-store-id cks-1234567890abcdef0
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS Key Management Service 개발자 안내서의 클라우드HSM 키 스토어를 참조하세요. AWS

예제 7: 외부 KMS 키 스토어에서 키를 생성하려면

다음 `create-key` 예제에서는 지정된 외부 KMS 키 스토어에 키를 생성합니다. 이 명령에는 `--custom-key-store-id`, `--origin`, `--xks-key-id` 파라미터가 필요합니다.

`--xks-key-id` 파라미터는 외부 키 관리자에 있는 기존 대칭 암호화 키의 ID를 지정합니다. 이 키는 KMS 키의 외부 키 구성 요소 역할을 합니다.`--origin`파라미터의 값은 여야 합니다`EXTERNAL_KEY_STORE`.`custom-key-store-id`파라미터는 외부 키 스토어 프록시에 연결된 외부 키 스토어를 식별해야 합니다.

```
aws kms create-key \
    --origin EXTERNAL_KEY_STORE \
    --custom-key-store-id cks-9876543210fedcba9 \
    --xks-key-id bb8562717f809024
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 AWS Key Management Service 개발자 안내서의 외부 키 저장소를 참조하세요.

- 자세한 API 내용은 명령 참조CreateKey의 섹션을 참조하세요. AWS CLI

### `decrypt`

다음 코드 예시에서는 `decrypt`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 대칭 KMS 키(Linux 및 macOS)를 사용하여 암호화된 메시지를 복호화하려면

다음 `decrypt` 명령 예제는 를 사용하여 데이터를 복호화하는 권장 방법을 보여줍니다 AWS CLI. 이 버전은 대칭 KMS 키로 데이터를 복호화하는 방법을 보여줍니다.

file.In `--ciphertext-blob` 파라미터 값을 암호 텍스트에 입력하고 `fileb://` 접두사를 사용합니다. 접두사는 CLI에 바이너리 파일에서 데이터를 읽도록 지시합니다. 파일이 현재 디렉터리에 없는 경우 파일의 전체 경로를 입력합니다. 파일에서 파라미터 값을 읽 AWS CLI는 방법에 대한 자세한 내용은 AWS Command Line Tool Blog의 Command Line Interface 사용 설명서 및 로컬 파일 파라미터 모범 사례<https://aws.amazon.com/blogs/developer/best-practices-for-local-

file-parameters/>의 <-https://docs.aws.amazon.com/cli/latest/userguide/cliusage-parameters-file.html> 파일에서 파라미터 AWS CLI로드를 참조하세요.암호텍스트를 해독하기 위한 KMS 키를 지정하세요. 대칭 KMS 키를 사용하여 복호화할 때 `--key-id` 파라미터가 필요하지 않습니다. AWS KMS 는 암호텍스트의 메타데이터에서 데이터를 암호화하는 데 사용된 KMS 키의 키 ID를 가져올 수 있습니다. AWS  하지만 사용하는 KMS 키를 지정하는 것이 항상 모범 사례입니다. 이 연습을 통해 의도한 KMS 키를 사용하고, 신뢰할 수 없는 KMS 키를 사용하여 암호 텍스트를 실수로 복호화하지 못하도록 할 수 있습니다. 일반 텍스트 출력을 텍스트 값으로 요청합니다. `--query` 파라미터는 출력에서 `Plaintext` 필드 값만 가져오CLI도록 에 지시합니다. `--output` 파라미터는 출력을 텍스트로 반환합니다. 일반 텍스트를 Base64로 디코딩하여 파일에 저장합니다. 다음 예시에서는 `Plaintext` 파라미터 값을 Base64 유틸리티에 파이프(|)로 구분하며 유틸리티가 이를 디코딩합니다. 그런 다음 디코딩된 출력을 `ExamplePlaintext` 파일로 리디렉션(>)합니다.

이 명령을 실행하기 전에 예제 키 ID를 AWS 계정의 유효한 키 ID로 바꿉니다.

```
aws kms decrypt \
    --ciphertext-blob fileb://ExampleEncryptedFile \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --output text \
    --query Plaintext | base64 \
    --decode > ExamplePlaintextFile
```

이 명령은 출력을 생성하지 않습니다. `decrypt` 명령의 출력은 base64로 디코딩되어 파일에 저장됩니다.

자세한 내용은 AWS 키 관리 서비스 API 참조의 암호 해독을 참조하세요.

예제 2: 대칭 KMS 키를 사용하여 암호화된 메시지를 복호화하려면(Windows 명령 프롬프트)

다음 예시는 `certutil` 유틸리티를 사용하여 일반 텍스트 데이터를 base64로 디코딩한다는 점을 제외하면 이전 예시와 동일합니다. 이 프로시저에는 다음 예시와 같이 두 개의 명령이 필요합니다.

이 명령을 실행하기 전에 예제 키 ID를 AWS 계정의 유효한 키 ID로 바꿉니다.

```
aws kms decrypt ^
    --ciphertext-blob fileb://ExampleEncryptedFile ^
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab ^
    --output text ^
    --query Plaintext > ExamplePlaintextFile.base64
```

`certutil` 명령을 실행합니다.

```
certutil -decode ExamplePlaintextFile.base64 ExamplePlaintextFile
```

출력:

```
Input Length = 18
Output Length = 12
CertUtil: -decode command completed successfully.
```

자세한 내용은 AWS 키 관리 서비스 API 참조의 암호 해독을 참조하세요.

예제 3: 비대칭 KMS 키(Linux 및 macOS)로 암호화된 메시지를 복호화하려면

다음 decrypt 명령 예제는 RSA 비대칭 KMS 키로 암호화된 데이터를 복호화하는 방법을 보여줍니다.

비대칭 KMS 키를 사용하는 경우 일반 텍스트를 암호화하는 데 사용되는 알고리즘을 지정하는 encryption-algorithm 파라미터가 필요합니다.

이 명령을 실행하기 전에 예제 키 ID를 AWS 계정의 유효한 키 ID로 바꿉니다.

```
aws kms decrypt \
    --ciphertext-blob fileb://ExampleEncryptedFile \
    --key-id 0987dcba-09fe-87dc-65ba-ab0987654321 \
    --encryption-algorithm RSAES_OAEP_SHA_256 \
    --output text \
    --query Plaintext | base64 \
    --decode > ExamplePlaintextFile
```

이 명령은 출력을 생성하지 않습니다. decrypt 명령의 출력은 base64로 디코딩되어 파일에 저장됩니다.

자세한 내용은 Key AWS Management Service 개발자 안내서의 에서 비대칭 AWS KMS 키를 참조하세요.

- API 자세한 내용은 AWS CLI 명령 참조의 복호화를 참조하세요.

## delete-alias

다음 코드 예시에서는 delete-alias을 사용하는 방법을 보여 줍니다.

AWS CLI

별칭을 AWS KMS 삭제하려면

다음 delete-alias 예시에서는 alias/example-alias 별칭을 삭제합니다. 별칭 이름은 alias/로 시작해야 합니다.

```
aws kms delete-alias \
    --alias-name alias/example-alias
```

이 명령은 출력을 생성하지 않습니다. 별칭을 찾으려면 list-aliases 명령을 사용하세요.

자세한 내용은 AWS Key Management Service 개발자 안내서의 별칭 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteAlias의 섹션을 참조하세요. AWS CLI

### delete-custom-key-store

다음 코드 예시에서는 delete-custom-key-store을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 키 스토어를 삭제하려면

다음 delete-custom-key-store 예제에서는 지정된 사용자 지정 키 스토어를 삭제합니다.

AWS 클라우드HSM 키 스토어를 삭제해도 연결된 클라우드HSM 클러스터에는 영향을 미치지 않습니다. 외부 키 스토어를 삭제해도 연결된 외부 키 스토어 프록시, 외부 키 관리자 또는 외부 키에는 영향을 미치지 않습니다.

NOTE: 사용자 지정 키 스토어를 삭제하려면 먼저 사용자 지정 KMS 키 스토어의 모든 키 삭제를 예약한 다음 해당 KMS 키가 삭제될 때까지 기다려야 합니다. 그런 다음 사용자 지정 키 스토어의 연결을 해제해야 합니다. 사용자 지정 KMS 키 스토어에서 키를 찾는 방법에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 AWS 클라우드HSM 키 스토어 삭제(API)를 참조하세요.

```
delete-custom-key-store \
    --custom-key-store-id cks-1234567890abcdef0
```

이 명령은 출력을 반환하지 않습니다. 사용자 지정 키 스토어가 삭제되었는지 확인하려면 describe-custom-key-stores 명령을 사용합니다.

AWS 클라우드HSM 키 스토어 삭제에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 AWS 클라우드HSM 키 스토어 삭제를 참조하세요.

외부 키 스토어 삭제에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 외부 키 스토어 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteCustomKeyStore의 섹션을 참조하세요. AWS CLI

### `delete-imported-key-material`

다음 코드 예시에서는 `delete-imported-key-material`을 사용하는 방법을 보여 줍니다.

AWS CLI

KMS 키에서 가져온 키 구성 요소를 삭제하려면

다음 `delete-imported-key-material` 예제에서는 키로 가져온 KMS 키 구성 요소를 삭제합니다.

```
aws kms delete-imported-key-material \
   --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

이 명령은 출력을 생성하지 않습니다. 키 구성 요소가 삭제되었는지 확인하려면 `describe-key` 명령을 사용하여 `PendingImport` 또는 의 키 상태를 찾습니다`PendingDeletion`.

자세한 내용은 Key AWS Management Service 개발자 안내서의 가져온 키 자료 삭제<https://docs.aws.amazon.com/kms/latest/developerguide/importing-keys-delete-key-material.html>를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteImportedKeyMaterial의 섹션을 참조하세요. AWS CLI

### `derive-shared-secret`

다음 코드 예시에서는 `derive-shared-secret`을 사용하는 방법을 보여 줍니다.

AWS CLI

공유 보안 암호를 파생하려면

다음 `derive-shared-secret` 예제에서는 키 계약 알고리즘을 사용하여 공유 보안 암호를 도출합니다.

를 호출하려면 KeyUsage 값이 인 비대칭 NIST권장 타원 곡선(ECC) 또는 SM2 (중국 리전만 해당) KMS 키 페어KEY_AGREEMENT를 사용해야 합니다 DeriveSharedSecret.

```
aws kms derive-shared-secret \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --key-agreement-algorithm ECDH \
    --public-
key "MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAvH3Yj0wbkLEpUl95Cv1cJVjsVNSjwGq3tCLnzXfhVwV
```

출력:

```
{
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "SharedSecret": "MEYCIQCKZLWyTk5runarx6XiAkU9gv3lbwPO/pHa
+DXFehzdDwIhANwpsIV2g/9SPWLLsF6p/hiSskuIXMTRwqrMdVKWTMHG",
    "KeyAgreementAlgorithm": "ECDH",
    "KeyOrigin": "AWS_KMS"
}
```

자세한 내용은 키 관리 서비스 참조DeriveSharedSecret의 섹션을 참조하세요. AWS API

- 자세한 API 내용은 명령 참조DeriveSharedSecret의 섹션을 참조하세요. AWS CLI

### describe-custom-key-stores

다음 코드 예시에서는 describe-custom-key-stores을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: AWS CloudHSM 키 스토어에 대한 세부 정보를 가져오려면

다음 describe-custom-key-store 예제에서는 지정된 AWS CloudHSM 키 스토어에 대한 세부 정보를 표시합니다. 명령은 모든 유형의 사용자 지정 키 스토어에서 동일하지만 출력은 키 스토어 유형과 다르고 외부 키 스토어의 경우 연결 옵션과 다릅니다.

기본적으로 이 명령은 계정 및 리전의 모든 사용자 지정 키 스토어에 대한 정보를 표시합니다. 특정 사용자 지정 키 스토어에 대한 정보를 표시하려면 custom-key-store-name 또는 custom-key-store-id 파라미터를 사용합니다.

```
aws kms describe-custom-key-stores \
```

```
    --custom-key-store-name ExampleCloudHSMKeyStore
```

이 명령의 출력에는 연결 상태()를 포함하여 AWS CloudHSM 키 스토어에 대한 유용한 세부 정보가 포함됩니다ConnectionState. 연결 상태가 인 경우 출력FAILED에는 문제를 설명하는 ConnectionErrorCode 필드가 포함됩니다.

출력:

```
{
    "CustomKeyStores": [
        {
            "CloudHsmClusterId": "cluster-1a23b4cdefg",
            "ConnectionState": "CONNECTED",
            "CreationDate": "2022-04-05T14:04:55-07:00",
            "CustomKeyStoreId": "cks-1234567890abcdef0",
            "CustomKeyStoreName": "ExampleExternalKeyStore",
            "TrustAnchorCertificate": "<certificate appears here>"
        }
    ]
}
```

자세한 내용은 Key Management Service 개발자 안내서의 AWS 클라우드HSM 키 스토어 보기를 참조하세요. AWS

예제 2: 퍼블릭 엔드포인트 연결이 있는 외부 키 스토어에 대한 세부 정보를 가져오려면

다음 describe-custom-key-store 예제에서는 지정된 외부 키 스토어에 대한 세부 정보를 표시합니다. 명령은 모든 유형의 사용자 지정 키 스토어에서 동일하지만 출력은 키 스토어 유형과 다르고 외부 키 스토어의 경우 연결 옵션과 다릅니다.

기본적으로 이 명령은 계정 및 리전의 모든 사용자 지정 키 스토어에 대한 정보를 표시합니다. 특정 사용자 지정 키 스토어에 대한 정보를 표시하려면 custom-key-store-name 또는 custom-key-store-id 파라미터를 사용합니다.

```
aws kms describe-custom-key-stores \
    --custom-key-store-id cks-9876543210fedcba9
```

이 명령의 출력에는 연결 상태()를 포함하여 외부 키 스토어에 대한 유용한 세부 정보가 포함됩니다ConnectionState. 연결 상태가 인 경우 출력FAILED에는 문제를 설명하는 ConnectionErrorCode 필드가 포함됩니다.

출력:

```
{
    "CustomKeyStores": [
        {
            "CustomKeyStoreId": "cks-9876543210fedcba9",
            "CustomKeyStoreName": "ExampleXKS",
            "ConnectionState": "CONNECTED",
            "CreationDate": "2022-12-02T07:48:55-07:00",
            "CustomKeyStoreType": "EXTERNAL_KEY_STORE",
            "XksProxyConfiguration": {
                "AccessKeyId": "ABCDE12345670EXAMPLE",
                "Connectivity": "PUBLIC_ENDPOINT",
                "UriEndpoint": "https://myproxy.xks.example.com",
                "UriPath": "/example-prefix/kms/xks/v1"
            }
        }
    ]
}
```

자세한 내용은 Key Management Service 개발자 안내서의 외부 키 스토어 보기를 참조하세요. AWS

예제 3: VPC 엔드포인트 서비스 연결이 있는 외부 키 스토어에 대한 세부 정보를 가져오려면

다음 `describe-custom-key-store` 예제에서는 지정된 외부 키 스토어에 대한 세부 정보를 표시합니다. 명령은 모든 유형의 사용자 지정 키 스토어에서 동일하지만 출력은 키 스토어 유형과 다르고 외부 키 스토어의 경우 연결 옵션과 다릅니다.

기본적으로 이 명령은 계정 및 리전의 모든 사용자 지정 키 스토어에 대한 정보를 표시합니다. 특정 사용자 지정 키 스토어에 대한 정보를 표시하려면 `custom-key-store-name` 또는 `custom-key-store-id` 파라미터를 사용합니다.

```
aws kms describe-custom-key-stores \
    --custom-key-store-id cks-2234567890abcdef0
```

이 명령의 출력에는 연결 상태()를 포함하여 외부 키 스토어에 대한 유용한 세부 정보가 포함됩니다`ConnectionState`. 연결 상태가 인 경우 출력`FAILED`에는 문제를 설명하는 `ConnectionErrorCode` 필드가 포함됩니다.

출력:

```
{
    "CustomKeyStores": [
        {
            "CustomKeyStoreId": "cks-3234567890abcdef0",
            "CustomKeyStoreName": "ExampleVPCExternalKeyStore",
            "ConnectionState": "CONNECTED",
            "CreationDate": "2022-12-22T07:48:55-07:00",
            "CustomKeyStoreType": "EXTERNAL_KEY_STORE",
            "XksProxyConfiguration": {
                "AccessKeyId": "ABCDE12345670EXAMPLE",
                "Connectivity": "VPC_ENDPOINT_SERVICE",
                "UriEndpoint": "https://myproxy-private.xks.example.com",
                "UriPath": "/kms/xks/v1",
                "VpcEndpointServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-
example1"
            }
        }
    ]
}
```

자세한 내용은 Key Management Service 개발자 안내서의 외부 키 스토어 보기를 참조하세요. AWS

- 자세한 API 내용은 명령 참조DescribeCustomKeyStores의 섹션을 참조하세요. AWS CLI

### `describe-key`

다음 코드 예시에서는 `describe-key`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: KMS 키에 대한 자세한 정보를 찾으려면

다음 `describe-key` 예제에서는 예제 계정 및 리전에서 Amazon S3의 AWS 관리형 키에 대한 자세한 정보를 가져옵니다. 이 명령을 사용하여 AWS 관리형 키 및 고객 관리형 키에 대한 세부 정보를 찾을 수 있습니다.

KMS 키를 지정하려면 `key-id` 파라미터를 사용합니다. 이 예제에서는 별칭 이름 값을 사용하지만 이 명령ARN에서 키 ID, 키 ARN, 별칭 이름 또는 별칭을 사용할 수 있습니다.

```
aws kms describe-key \
    --key-id alias/aws/s3
```

출력:

```
{
    "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"
        ]
    }
}
```

자세한 내용은 AWS Key Management Service 개발자 안내서의 키 보기를 참조하세요.

예제 2: RSA 비대칭 KMS 키에 대한 세부 정보를 가져오려면

다음 `describe-key` 예제에서는 서명 및 확인에 사용되는 비대칭 RSA KMS 키에 대한 자세한 정보를 가져옵니다.

```
aws kms describe-key \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

출력:

```
{
    "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"
        ]
    }
}
```

예 3: 다중 리전 복제본 키에 대한 세부 정보를 가져오는 방법

다음 describe-key 예시에서는 다중 리전 복제본 키에 대한 메타데이터를 가져옵니다. 이 다중 리전 키는 대칭 암호화 키입니다. 모든 다중 리전 키에 대한 describe-key 명령 출력은 프라이머리 키와 모든 해당 복제본에 대한 정보를 반환합니다.

```
aws kms describe-key \
    --key-id arn:aws:kms:ap-northeast-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab
```

출력:

```
{
    "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"
                }
            ]
        }
    }
}
```

예제 4: HMAC KMS 키에 대한 세부 정보를 가져오려면

다음 `describe-key` 예제에서는 HMAC KMS 키에 대한 자세한 정보를 가져옵니다.

```
aws kms describe-key \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

출력:

```
{
    "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
    }
}
```

- 자세한 API 내용은 명령 참조DescribeKey의 섹션을 참조하세요. AWS CLI

### disable-key-rotation

다음 코드 예시에서는 disable-key-rotation을 사용하는 방법을 보여 줍니다.

AWS CLI

KMS 키의 자동 교체를 비활성화하려면

다음 disable-key-rotation 예제에서는 고객 관리형 KMS 키의 자동 교체를 비활성화합니다. 자동 교체를 다시 활성화하려면 enable-key-rotation 명령을 사용합니다.

```
aws kms disable-key-rotation \
    --key-id arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
```

이 명령은 출력을 생성하지 않습니다. KMS 키에 자동 교체가 비활성화되어 있는지 확인하려면 get-key-rotation-status 명령을 사용합니다.

자세한 내용은 Key AWS Management Service 개발자 안내서의 키 교체를 참조하세요.

- 자세한 API 내용은 명령 참조DisableKeyRotation의 섹션을 참조하세요. AWS CLI

### disable-key

다음 코드 예시에서는 disable-key을 사용하는 방법을 보여 줍니다.

AWS CLI

KMS 키를 일시적으로 비활성화하려면

다음 예제에서는 disable-key 명령을 사용하여 고객 관리형 KMS 키를 비활성화합니다. KMS 키를 다시 활성화하려면 enable-key 명령을 사용합니다.

```
aws kms disable-key \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Key Management Service 개발자 안내서의 키 활성화 및 비활성화를 참조하세요.

- 자세한 API 내용은 명령 참조DisableKey의 섹션을 참조하세요. AWS CLI

### disconnect-custom-key-store

다음 코드 예시에서는 disconnect-custom-key-store을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 키 스토어 연결을 해제하려면

다음 disconnect-custom-key-store 예제에서는 AWS 클라우드HSM 클러스터에서 사용자 지정 키 스토어의 연결을 해제합니다. 문제를 해결하거나 설정을 업데이트하거나 키 스토어의 KMS 키가 암호화 작업에 사용되지 않도록 키 스토어의 연결을 해제할 수 있습니다.

이 명령은 AWS CloudHSM 키 스토어 및 외부 키 스토어를 포함한 모든 사용자 지정 키 스토어에서 동일합니다.

이 명령을 실행하기 앞서 예제에 나온 사용자 지정 키 스토어 ID를 유효한 ID로 대체합니다.

```
$ aws kms disconnect-custom-key-store \
```

```
    --custom-key-store-id cks-1234567890abcdef0
```

이 명령은 출력을 생성하지 않습니다. 명령이 유효한지 확인하고 `describe-custom-key-stores` 명령을 사용합니다.

AWS CloudHSM 키 스토어 연결 해제에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 AWS CloudHSM 키 스토어 연결 및 연결 해제를 참조하세요.

외부 키 스토어 연결 해제에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 외부 키 스토어 연결 및 연결 해제를 참조하세요.

- 자세한 API 내용은 명령 참조DisconnectCustomKeyStore의 섹션을 참조하세요. AWS CLI

### `enable-key-rotation`

다음 코드 예시에서는 `enable-key-rotation`을 사용하는 방법을 보여 줍니다.

AWS CLI

KMS 키의 자동 교체를 활성화하려면

다음 `enable-key-rotation` 예제에서는 180일의 교체 기간으로 고객 관리형 KMS 키를 자동으로 교체할 수 있습니다. KMS 키는 이 명령이 완료된 날짜로부터 1년(약 365일) 후에 교체되며 그 이후에는 매년 교체됩니다.

`--key-id` 파라미터는 KMS 키를 식별합니다. 이 예제에서는 키 ARN 값을 사용하지만 키 ID 또는 KMS 키ARN의 를 사용할 수 있습니다. `--rotation-period-in-days` 파라미터는 각 교체 날짜 사이의 일수를 지정합니다. 90~2560일 사이의 값을 지정합니다. 값을 지정하지 않으면 기본값은 365일입니다.

```
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
```

이 명령은 출력을 생성하지 않습니다. KMS 키가 활성화되어 있는지 확인하려면 `get-key-rotation-status` 명령을 사용합니다.

자세한 내용은 Key AWS Management Service 개발자 안내서의 키 교체를 참조하세요.

- 자세한 API 내용은 명령 참조EnableKeyRotation의 섹션을 참조하세요. AWS CLI

### enable-key

다음 코드 예시에서는 enable-key을 사용하는 방법을 보여 줍니다.

AWS CLI

KMS 키를 활성화하려면

다음 enable-key 예시에서는 고객 관리형 키를 활성화합니다. 이 명령과 같은 disable-key 명령을 사용하여 명령을 사용하여 일시적으로 비활성화한 KMS 키를 활성화할 수 있습니다. 삭제가 예약되고 삭제가 취소되었으므로 비활성화된 KMS 키를 활성화하는 데 사용할 수도 있습니다.

KMS 키를 지정하려면 key-id 파라미터를 사용합니다. 이 예제에서는 키 ID 값을 사용하지만 이 명령에서 키 ID 또는 키 ARN 값을 사용할 수 있습니다.

이 명령을 실행하기 전에 예시에 나온 키 ID를 유효한 키 핸들로 바꾸세요.

```
aws kms enable-key \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

이 명령은 출력을 생성하지 않습니다. KMS 키가 활성화되었는지 확인하려면 describe-key 명령을 사용합니다. describe-key 출력의 KeyState 및 Enabled 필드 값을 확인하세요.

자세한 내용은 AWS Key Management Service 개발자 안내서의 키 활성화 및 비활성화를 참조하세요.

- 자세한 API 내용은 명령 참조EnableKey의 섹션을 참조하세요. AWS CLI

### encrypt

다음 코드 예시에서는 encrypt을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: Linux 또는 macOS에서 파일 콘텐츠를 암호화하는 방법

다음 encrypt 명령은 를 사용하여 데이터를 암호화하는 권장 방법을 보여줍니다 AWS CLI.

```
aws kms encrypt \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --plaintext fileb://ExamplePlaintextFile \
    --output text \
    --query CiphertextBlob | base64 \
```

```
    --decode > ExampleEncryptedFile
```

이 명령은 여러 가지 작업을 수행합니다.

--plaintext 파라미터를 사용하여 암호화할 데이터를 표시합니다. 이 파라미터 값은 base64로 인코딩되어야 합니다.plaintext파라미터의 값은 base64로 인코딩되거나, 파일에서 바이너리 데이터를 읽도록 에 AWS CLI 지시하는 fileb:// 접두사를 사용해야 합니다. 파일이 현재 디렉터리에 없는 경우 파일의 전체 경로를 입력합니다. 예: fileb:///var/tmp/ExamplePlaintextFile 또는 fileb://C:\Temp\ExamplePlaintextFile. 파일에서 파라미터 값을 읽 AWS CLI는 방법에 대한 자세한 내용은 AWS 명령줄 인터페이스 사용 설명서의 파일에서 파라미터 로드 및 AWS 명령줄 도구 블로그의 로컬 파일 파라미터 모범 사례를 참조하세요. --output 및 --query 파라미터를 사용하여 명령의 출력을 제어합니다. 이러한 파라미터는 암호화된 데이터를 추출합니다. 를 암호 텍스트 라고 했습니다. 명령의 출력에서.출력 제어에 대한 자세한 내용은 AWS 명령줄 인터페이스 사용 설명서의 명령 출력 제어를 참조하세요. base64 유틸리티를 사용하여 추출된 출력을 바이너리 데이터로 디코딩합니다. 성공적인 encrypt 명령으로 반환되는 암호 텍스트는 base64로 인코딩된 텍스트입니다. 를 AWS CLI 사용하여 복호화하려면 먼저 이 텍스트를 복호화해야 합니다.이진 암호텍스트를 파일에 저장합니다. 명령의 마지막 부분(> ExampleEncryptedFile)은 이진 암호텍스트를 파일에 저장하여 복호화가 더 쉬워지도록 합니다. 를 AWS CLI 사용하여 데이터를 복호화하는 예제 명령은 복호화 예제를 참조하세요.

예제 2: AWS CLI를 사용하여 Windows에서 데이터 암호화

이 예시는 base64 대신 certutil 도구를 사용한다는 점을 제외하면 이전 예와 동일합니다. 이 절차에는 다음 예시와 같이 두 개의 명령이 필요합니다.

```
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
```

예제 3: 비대칭 KMS 키로 암호화

다음 encrypt 명령은 비대칭 KMS 키로 일반 텍스트를 암호화하는 방법을 보여줍니다. --encryption-algorithm 파라미터가 필요합니다. 모든 encrypt CLI 명령과 마찬가지로 plaintext 파라미터는 base64로 인코딩되거나 에 AWS CLI 파일에서 바이너리 데이터를 읽도록 지시하는 fileb:// 접두사를 사용해야 합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

- API 자세한 내용은 AWS CLI 명령 참조의 암호화를 참조하세요.

### generate-data-key-pair-without-plaintext

다음 코드 예시에서는 generate-data-key-pair-without-plaintext을 사용하는 방법을 보여 줍니다.

AWS CLI

ECC NIST P384 비대칭 데이터 키 페어를 생성하려면

다음 generate-data-key-pair-without-plaintext 예제에서는 외부에서 사용할 ECC NIST P384 키 페어를 요청합니다 AWS.

명령은 일반 텍스트 퍼블릭 키와 지정된 키로 암호화된 프라이빗 KMS 키의 사본을 반환합니다. 일반 텍스트 프라이빗 키는 반환되지 않습니다. 암호화된 프라이빗 키를 암호화된 데이터와 함께 안전하게 저장하고 를 호출 AWS KMS하여 프라이빗 키를 사용해야 할 때 복호화할 수 있습니다.

ECC NIST P384 비대칭 데이터 키 페어를 요청하려면 값이 인 key-pair-spec 파라미터를 사용합니다ECC_NIST_P384.

지정하는 KMS 키는 대칭 암호화 KMS 키, 즉 KeySpec 값이 인 KMS 키여야 합니다SYMMETRIC_DEFAULT.

NOTE: 이 예제의 출력에 있는 값은 표시를 위해 잘립니다.

```
aws kms generate-data-key-pair-without-plaintext \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --key-pair-spec ECC_NIST_P384
```

출력:

```
{
    "PrivateKeyCiphertextBlob": "AQIDAHi6LtupRpdKl2aJTzkK6FbhOtQkMlQJJH3PdtHvS/y
+hAFFxmiD134doUDzMGmfCEtcAAAHaTCCB2UGCSqGSIb3DQEHBqCCB1...",
    "PublicKey":
 "MIIBojANBgkqhkiG9w0BAQEFAAOCAY8AMIIBigKCAYEA3A3eGMyPrvSn7+LdlJE1oUoQV5HpEuHAVbdOyND
+NmYDH/mL1OSIEuLrcdZ5hrMH4pk83r40l...",
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyPairSpec": "ECC_NIST_P384"
}
```

PublicKey 및 는 base64 인코딩 형식으로 반환PrivateKeyCiphertextBlob됩니다.

자세한 내용은 Key Management Service 개발자 안내서의 데이터 키 페어AWS 를 참조하세요.

- 자세한 API 내용은 명령 참조GenerateDataKeyPairWithoutPlaintext의 섹션을 참조하세요. AWS CLI

### generate-data-key-pair

다음 코드 예시에서는 generate-data-key-pair을 사용하는 방법을 보여 줍니다.

AWS CLI

2048비트 RSA 비대칭 데이터 키 페어를 생성하려면

다음 generate-data-key-pair 예제에서는 외부에서 사용할 2048비트 RSA 비대칭 데이터 키 페어를 요청합니다 AWS. 명령은 즉시 사용하고 삭제할 수 있도록 일반 텍스트 퍼블릭 키와 일반 텍스트 프라이빗 키, 지정된 키로 암호화된 프라이빗 KMS 키의 사본을 반환합니다. 암호화된 프라이빗 키를 암호화된 데이터와 함께 안전하게 저장할 수 있습니다.

2048비트 RSA 비대칭 데이터 키 페어를 요청하려면 값이 인 key-pair-spec 파라미터를 사용합니다RSA_2048.

지정하는 KMS 키는 대칭 암호화 KMS 키, 즉 KeySpec 값이 인 KMS 키여야 합니다SYMMETRIC_DEFAULT.

NOTE: 이 예제의 출력에 있는 값은 표시를 위해 잘립니다.

```
aws kms generate-data-key-pair \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --key-pair-spec RSA_2048
```

출력:

```
{
    "PrivateKeyCiphertextBlob": "AQIDAHi6LtupRpdKl2aJTzkK6FbhOtQkMlQJJH3PdtHvS/y
+hAFFxmiD134doUDzMGmfCEtcAAAHaTCCB2UGCSqGSIb3DQEHBqCCB1...",
    "PrivateKeyPlaintext": "MIIG/
QIBADANBgkqhkiG9w0BAQEFAASCBucwggbjAgEAAoIBgQDcDd4YzI
+u9Kfv4t2UkTWhShBXkekS4cBVt07I0P42ZgMf+YvU5IgS4ut...",
    "PublicKey":
 "MIIBojANBgkqhkiG9w0BAQEFAAOCAY8AMIIBigKCAYEA3A3eGMyPrvSn7+LdlJE1oUoQV5HpEuHAVbdOyND
+NmYDH/mL1OSIEuLrcdZ5hrMH4pk83r40l...",
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyPairSpec": "RSA_2048"
}
```

PublicKey, PrivateKeyPlaintext및 는 base64 인코딩 형식으로 반환PrivateKeyCiphertextBlob됩니다.

자세한 내용은 Key Management Service 개발자 안내서의 데이터 키 페어AWS 를 참조하세요.

- 자세한 API 내용은 명령 참조GenerateDataKeyPair의 섹션을 참조하세요. AWS CLI

### generate-data-key-without-plaintext

다음 코드 예시에서는 generate-data-key-without-plaintext을 사용하는 방법을 보여 줍니다.

AWS CLI

일반 텍스트 키 없이 256비트 대칭 데이터 키를 생성하는 방법

다음 generate-data-key-without-plaintext 예시에서는 AWS외부에서 사용할 256비트 대칭 데이터 키의 암호화된 사본을 요청합니다. 사용할 준비가 되면 를 호출 AWS KMS하여 데이터 키를 복호화할 수 있습니다.

256비트 데이터 키를 요청하려면 값이 AES_256인 key-spec 파라미터를 사용하세요. 128비트 데이터 키를 요청하려면 값이 AES_128인 key-spec 파라미터를 사용하세요. 다른 모든 데이터 키 길이에는 number-of-bytes 파라미터를 사용하세요.

지정하는 KMS 키는 대칭 암호화 KMS 키, 즉 키 사양 값이 SYMMETRIC_인 KMS 키여야 합니다DEFAULT.

```
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
```

출력:

```
{
    "CiphertextBlob":
 "AQEDAHjRYf5WytIc0C857tFSnBaPn2F8DgfmThbJlGfR8P3WlwAAAH4wfAYJKoZIhvcNAQcGoG8wbQIBADBoBgkqhk
    "KeyId": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
}
```

CiphertextBlob(암호화된 데이터 키)은 base64로 인코딩된 형식으로 반환됩니다.

자세한 내용은 AWS Key Management Service 개발자 안내서의 데이터 키를 참조하세요.

- 자세한 API 내용은 명령 참조GenerateDataKeyWithoutPlaintext의 섹션을 참조하세요. AWS CLI

### generate-data-key

다음 코드 예시에서는 generate-data-key을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: 256비트 대칭 데이터 키를 생성하는 방법

다음 generate-data-key 예제에서는 외부에서 사용할 256비트 대칭 데이터 키를 요청합니다 AWS. 명령은 즉시 사용 및 삭제할 수 있는 일반 텍스트 데이터 키와 지정된 키로 암호화된 해당 데이터 KMS 키의 사본을 반환합니다. 암호화한 데이터 키를 암호화한 데이터와 함께 안전하게 저장할 수 있습니다.

256비트 데이터 키를 요청하려면 값이 AES_256인 key-spec 파라미터를 사용하세요. 128비트 데이터 키를 요청하려면 값이 AES_128인 key-spec 파라미터를 사용하세요. 다른 모든 데이터 키 길이에는 number-of-bytes 파라미터를 사용하세요.

지정하는 KMS 키는 대칭 암호화 KMS 키, 즉 키 사양 값이 SYMMETRIC_인 KMS 키여야 합니다 DEFAULT.

```
aws kms generate-data-key \
```

```
    --key-id alias/ExampleAlias \
    --key-spec AES_256
```

출력:

```
{
    "Plaintext": "VdzKNHGzUAzJeRBVY+uUmofUGGiDzyB3+i9fVkh3piw=",
    "KeyId": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "CiphertextBlob":
 "AQEDAHjRYf5WytIc0C857tFSnBaPn2F8DgfmThbJlGfR8P3WlwAAAH4wfAYJKoZIhvcNAQcGoG8wbQIBADBoBgkqhk
+YdhV8MrkBQPeac0ReRVNDt9qleAt+SHgIRF8P0H+7U="
}
```

`Plaintext`(일반 텍스트 데이터 키) 및 `CiphertextBlob`(암호화된 데이터 키)은 base64로 인코딩된 형식으로 반환됩니다.

자세한 내용은 Key AWS Management Service 개발자 안내서의 데이터 키 <https://docs.aws.amazon.com/kms/latest/developerguide/concepts.html#data-keys를 참조하세요.

예 2: 512비트 대칭 데이터 키를 생성하는 방법

다음 `generate-data-key` 예시에서는 암호화 및 복호화를 위한 512비트 대칭 데이터 키를 요청합니다. 명령은 즉시 사용 및 삭제할 수 있는 일반 텍스트 데이터 키와 지정된 키로 암호화된 해당 데이터 KMS 키의 사본을 반환합니다. 암호화한 데이터 키를 암호화한 데이터와 함께 안전하게 저장할 수 있습니다.

128비트 또는 256비트가 아닌 키 길이를 요청하려면 `number-of-bytes` 파라미터를 사용하세요. 512비트 데이터 키를 요청하기 위해 다음 예시에서는 값이 64(바이트)인 `number-of-bytes` 파라미터를 사용합니다.

지정하는 KMS 키는 대칭 암호화 KMS 키, 즉 키 사양 값이 SYMMETRIC_인 KMS 키여야 합니다 DEFAULT.

NOTE: 이 예제의 출력에 있는 값은 표시를 위해 잘립니다.

```
aws kms generate-data-key \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --number-of-bytes 64
```

출력:

```
{
    "CiphertextBlob": "AQIBAHi6LtupRpdKl2aJTzkK6FbhOtQkMlQJJH3PdtHvS/y+hAEnX/
QQNmMwDfg2korNMEc8AAACaDCCAmQGCSqGSIb3DQEHBqCCAlUwggJRAgEAMIICSgYJKoZ...",
    "Plaintext": "ty8Lr0Bk6OF07M2BWt6qbFdNB
+G00ZLtf5MSEb4al3R2UKWGOp06njAwy2n72VRm2m7z/
Pm9Wpbvttz6a4lSo9hgPvKhZ5y6RTm4OovEXiVfBveyX3DQxDzRSwbKDPk/...",
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
}
```

`Plaintext`(일반 텍스트 데이터 키) 및 `CiphertextBlob`(암호화된 데이터 키)은 base64로 인코딩된 형식으로 반환됩니다.

자세한 내용은 Key AWS Management Service 개발자 안내서의 데이터 키 <https://docs.aws.amazon.com/kms/latest/developerguide/concepts.html#data-keys를 참조하세요.

- 자세한 API 내용은 명령 참조GenerateDataKey의 섹션을 참조하세요. AWS CLI

**`generate-random`**

다음 코드 예시에서는 `generate-random`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 256비트 랜덤 바이트 문자열 생성(Linux 또는 macOs)

다음 `generate-random` 예시에서는 256비트(32바이트), base64로 인코딩된 무작위 바이트 문자열을 생성합니다. 이 예시에서는 바이트 문자열을 디코딩하여 무작위 파일에 저장합니다.

이 명령을 실행할 때는 `number-of-bytes` 파라미터를 사용하여 무작위 값의 길이를 바이트 단위로 지정해야 합니다.

이 명령을 실행할 때는 KMS 키를 지정하지 않습니다. 임의 바이트 문자열은 KMS 키와 관련이 없습니다.

기본적으로 AWS KMS는 임의 번호를 생성합니다. 하지만 사용자 지정 키 스토어<https://docs.aws.amazon.com/kms/latest/developerguide/custom-key-store-overview.html>를 지정하면 사용자 지정 키 스토어와 연결된 AWS 클라우드HSM 클러스터에 무작위 바이트 문자열이 생성됩니다.

이 예시에서는 다음 파라미터와 값을 사용합니다.

값이 인 필수 --number-of-bytes 파라미터를 사용하여 32바이트(256비트)를 32 요청합니다. string.It 값이 인 --output 파라미터를 사용하여 가 AWS CLI 출력을 텍스트로 반환하도록 text 지시합니다. 대신 JSON입니다. --query parameter를 사용하여 base64 response.It 파이프( | ) 에서 Plaintext 유틸리티에 대한 명령의 출력을 추출합니다. 는 추출된 output.It 리디렉션 연산자 ( > )를 사용하여 디코딩된 바이트 문자열을 ExampleRandom file.It 리디렉션 연산자( > )를 사용하여 바이너리 암호 텍스트를 파일에 저장합니다.

```
aws kms generate-random \
    --number-of-bytes 32 \
    --output text \
    --query Plaintext | base64 --decode > ExampleRandom
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 키 관리 서비스 참조GenerateRandom의 섹션을 참조하세요. AWS API

예 2: 256비트 무작위 수를 생성하는 방법(Windows 명령 프롬프트)

다음 예시에서는 generate-random 명령을 사용하여 256비트(32바이트), base64로 인코딩된 무작위 바이트 문자열을 생성합니다. 이 예시에서는 바이트 문자열을 디코딩하여 무작위 파일에 저장합니다. 이 예시는 Windows의 certutil 유틸리티를 사용하여 무작위 바이트 문자열을 base64로 디코딩한 다음 파일에 저장한다는 점을 제외하면 이전 예와 동일합니다.

먼저 base64로 인코딩된 무작위 바이트 문자열을 생성하여 임시 파일 ExampleRandom.base64에 저장합니다.

```
aws kms generate-random \
    --number-of-bytes 32 \
    --output text \
    --query Plaintext > ExampleRandom.base64
```

generate-random 명령의 출력이 파일에 저장되기 때문에 이 예시에서는 출력이 생성되지 않습니다.

이제 certutil -decode 명령을 사용하여 ExampleRandom.base64 파일에서 base64로 인코딩된 바이트 문자열을 디코딩합니다. 그런 다음 디코딩된 바이트 문자열을 ExampleRandom 파일에 저장합니다.

```
certutil -decode ExampleRandom.base64 ExampleRandom
```

출력:

```
Input Length = 18
Output Length = 12
CertUtil: -decode command completed successfully.
```

자세한 내용은 키 관리 서비스 참조GenerateRandom의 섹션을 참조하세요. AWS API

- 자세한 API 내용은 명령 참조GenerateRandom의 섹션을 참조하세요. AWS CLI

### get-key-policy

다음 코드 예시에서는 get-key-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

한 키에서 다른 KMS 키로 KMS 키 정책을 복사하려면

다음 get-key-policy 예제는 한 키에서 KMS 키 정책을 가져와 텍스트 파일에 저장합니다. 그런 다음 텍스트 파일을 정책 입력으로 사용하여 다른 KMS 키의 정책을 대체합니다.

의 --policy 파라미터에는 문자열이 put-key-policy 필요하므로 --output text 옵션을 사용하여 출력을 대신 텍스트 문자열로 반환해야 합니다JSON.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 참조PutKeyPolicy의 섹션을 참조하세요. AWS KMS API

- 자세한 API 내용은 명령 참조GetKeyPolicy의 섹션을 참조하세요. AWS CLI

### get-key-rotation-status

다음 코드 예시에서는 get-key-rotation-status을 사용하는 방법을 보여 줍니다.

AWS CLI

KMS 키의 교체 상태를 검색합니다.

다음 get-key-rotation-status 예제에서는 자동 교체 활성화 여부, 교체 기간 및 다음 예정된 교체 날짜를 포함하여 지정된 KMS 키의 교체 상태에 대한 정보를 반환합니다. 고객 관리형 KMS 키 및 AWS 관리형 KMS 키에서 이 명령을 사용할 수 있습니다. 그러나 모든 AWS 관리형 KMS 키는 매년 자동으로 교체됩니다.

```
aws kms get-key-rotation-status \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

출력:

```
{
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyRotationEnabled": true,
    "NextRotationDate": "2024-02-14T18:14:33.587000+00:00",
    "RotationPeriodInDays": 365
}
```

자세한 내용은 Key Management Service 개발자 안내서의 키 교체를 참조하세요. AWS

- 자세한 API 내용은 명령 참조GetKeyRotationStatus의 섹션을 참조하세요. AWS CLI

### get-parameters-for-import

다음 코드 예시에서는 get-parameters-for-import을 사용하는 방법을 보여 줍니다.

AWS CLI

키 구성 요소를 KMS 키로 가져오는 데 필요한 항목을 가져오려면

다음 get-parameters-for-import 예제에서는 키 구성 요소를 키로 가져오는 데 필요한 퍼블릭 키와 가져오기 토큰을 가져옵니다KMS. import-key-material 명령을 사용할 때는 동일한 get-parameters-for-import 명령으로 반환된 퍼블릭 키로 암호화된 가져오기 토큰과 키 구성 요소를 사용해야 합니다. 또한 이 명령에서 지정하는 래핑 알고리즘은 퍼블릭 키로 키 구성 요소를 암호화하는 데 사용하는 알고리즘이어야 합니다.

KMS 키를 지정하려면 key-id 파라미터를 사용합니다. 이 예제에서는 키 ID를 사용하지만 이 명령 ARN에서 키 ID 또는 키를 사용할 수 있습니다.

```
aws kms get-parameters-for-import \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --wrapping-algorithm RSAES_OAEP_SHA_256 \
    --wrapping-key-spec RSA_2048
```

출력:

```
{
    "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
}
```

자세한 내용은 AWS Key Management Service 개발자 안내서의 퍼블릭 키 다운로드 및 토큰 가져오기를 참조하세요.

- 자세한 API 내용은 명령 참조GetParametersForImport의 섹션을 참조하세요. AWS CLI

### get-public-key

다음 코드 예시에서는 get-public-key을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 비대칭 키의 퍼블릭 KMS 키를 다운로드하려면

다음 get-public-key 예제에서는 비대칭 키의 퍼블릭 KMS 키를 다운로드합니다.

퍼블릭 키를 반환하는 것 외에도 출력에는 키 사용 및 지원되는 암호화 알고리즘을 AWS KMS포함하여 외부에서 퍼블릭 키를 안전하게 사용하는 데 필요한 정보가 포함됩니다.

```
aws kms get-public-key \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

출력:

```
{
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
```

```
    "PublicKey": "jANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAl5epvg1/
QtJhxSi2g9SDEVg8QV/...",
    "CustomerMasterKeySpec": "RSA_4096",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "EncryptionAlgorithms": [
        "RSAES_OAEP_SHA_1",
        "RSAES_OAEP_SHA_256"
    ]
}
```

의 비대칭 KMS 키 사용에 대한 자세한 내용은 AWS 키 관리 서비스 API 참조의 대칭 및 비대칭 키 사용을 AWS KMS참조하세요. https://docs.aws.amazon.com/kms/latest/developerguide/symmetric-asymmetric.html

예제 2: 퍼블릭 키를 DER 형식으로 변환(Linux 및 macOS)

다음 `get-public-key` 예제에서는 비대칭 키의 퍼블릭 KMS 키를 다운로드하여 DER 파일에 저장합니다.

에서 `get-public-key` 명령을 사용하면 Base64-encoded DER인코딩 인 인코딩된 X.509 퍼블릭 키를 AWS CLI반환합니다. 이 예제에서는 `PublicKey` 속성 값을 텍스트로 가져옵니다. Base64-decodesPublicKey하고 `public_key.der` 파일에 저장합니다. `output` 파라미터는 출력을 대신 텍스트로 반환합니다JSON. `--query` 파라미터는 `PublicKey` 외부에서 퍼블릭 키를 안전하게 사용하는 데 필요한 속성이 아닌 속성만 가져옵니다 AWS KMS.

이 명령을 실행하기 전에 예제 키 ID를 AWS 계정의 유효한 키 ID로 바꿉니다.

```
aws kms get-public-key \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --output text \
    --query PublicKey | base64 --decode > public_key.der
```

이 명령은 출력을 생성하지 않습니다.

의 비대칭 KMS 키 사용에 대한 자세한 AWS KMS내용은 AWS Key Management Service API 참조의 대칭 및 비대칭 키 사용을 참조하세요.

- 자세한 API 내용은 명령 참조GetPublicKey의 섹션을 참조하세요. AWS CLI

**`import-key-material`**

다음 코드 예시에서는 `import-key-material`을 사용하는 방법을 보여 줍니다.

AWS CLI

키 구성 요소를 KMS 키로 가져오려면

다음 import-key-material 예제에서는 키 구성 요소를 키 구성 요소 없이 생성된 KMS 키에 업로드합니다. 키의 KMS 키 상태는 여야 합니다PendingImport.

이 명령은 get-parameters-for-import 명령이 반환한 퍼블릭 키로 암호화된 키 구성 요소를 사용합니다. 또한 동일한 get-parameters-for-import 명령의 가져오기 토큰을 사용합니다.

expiration-model 파라미터는 키 구성 요소가 valid-to 파라미터에 지정된 날짜 및 시간에 자동으로 만료됨을 나타냅니다. 키 구성 요소가 만료 AWS KMS되면 가 키 구성 요소를 삭제Pending import하고 키의 KMS 키 상태가 로 변경되며 KMS 키를 사용할 수 없게 됩니다. KMS 키를 복원하려면 동일한 키 구성 요소를 다시 가져와야 합니다. 다른 키 구성 요소를 사용하려면 새 KMS 키를 생성해야 합니다.

이 명령을 실행하기 전에 예제 키 ID를 AWS 계정의 유효한 키 ID 또는 키로 바ARN꿉니다.

```
aws kms import-key-material \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --encrypted-key-material fileb://EncryptedKeyMaterial.bin \
    --import-token fileb://ImportToken.bin \
    --expiration-model KEY_MATERIAL_EXPIRES \
    --valid-to 2021-09-21T19:00:00Z
```

이 명령은 출력을 생성하지 않습니다.

키 구성 요소 가져오기에 대한 자세한 내용은 키 AWS 관리 서비스 개발자 안내서의 키 구성 요소 가져오기를 참조하세요.

- 자세한 API 내용은 명령 참조ImportKeyMaterial의 섹션을 참조하세요. AWS CLI

## list-aliases

다음 코드 예시에서는 list-aliases을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: AWS 계정 및 리전의 모든 별칭을 나열하려면

다음 예제에서는 list-aliases 명령을 사용하여 AWS 계정의 기본 리전에 있는 모든 별칭을 나열합니다. 출력에는 AWS 관리형 KMS 키 및 고객 관리형 KMS 키와 연결된 별칭이 포함됩니다.

```
aws kms list-aliases
```

출력:

```
{
    "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"
        },
        ...
    ]
}
```

예제 2: 특정 KMS 키의 모든 별칭을 나열하려면

다음 예제에서는 `list-aliases` 명령과 해당 `key-id` 파라미터를 사용하여 특정 KMS 키와 연결된 모든 별칭을 나열합니다.

각 별칭은 하나의 KMS 키에만 연결되지만 KMS 키에는 여러 개의 별칭이 있을 수 있습니다. 콘솔에는 각 KMS 키에 AWS KMS 대해 하나의 별칭만 나열되므로 이 명령은 매우 유용합니다. KMS 키의 모든 별칭을 찾으려면 `list-aliases` 명령을 사용해야 합니다.

이 예제에서는 `--key-id` 파라미터에 대한 KMS 키의 키 ID를 사용하지만 이 명령ARN에서 키 ID, 키 ARN, 별칭 이름 또는 별칭을 사용할 수 있습니다.

```
aws kms list-aliases --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS Key Management Service 개발자 안내서의 별칭으로 작업을 참조하세요.

- 자세한 API 내용은 명령 참조ListAliases의 섹션을 참조하세요. AWS CLI

### `list-grants`

다음 코드 예시에서는 `list-grants`을 사용하는 방법을 보여 줍니다.

AWS CLI

키에서 권한 부여를 AWS KMS 보려면

다음 `list-grants` 예제에서는 계정의 Amazon DynamoDB에 지정된 AWS 관리형 KMS 키에 대한 모든 권한을 표시합니다. 이 권한 부여를 통해 DynamoDB는 사용자를 대신하여 KMS 키를 사용하여 디스크에 쓰기 전에 DynamoDB 테이블을 암호화할 수 있습니다. 이 명령과 같은 명령을 사용하여 AWS 계정 및 리전의 AWS 관리형 KMS 키 및 고객 관리형 KMS 키에 대한 권한 부여를 볼 수 있습니다.

이 명령은 키 ID가 있는 `key-id` 파라미터를 사용하여 KMS 키를 식별합니다. 키 ID 또는 키를 사용하여 KMS 키를 ARN 식별할 수 있습니다. ARN AWS 관리형 키의 KMS 키 ID 또는 키를 가져오려면 `list-keys` 또는 `list-aliases` 명령을 사용합니다.

```
aws kms list-grants \
```

```
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

출력에 따르면 권한 부여는 암호화 작업에 KMS 키를 사용할 수 있는 권한을 Amazon DynamoDB에 부여하고 KMS 키에 대한 세부 정보를 보고(DescribeKey) 권한 부여를 사용 중지할 수 있는 권한을 Amazon DynamoDB에 부여합니다(RetireGrant). EncryptionContextSubset 제약 조건은 이러한 권한을 지정된 암호화 컨텍스트 페어를 포함하는 요청으로 제한합니다. 따라서 권한 부여의 권한은 지정된 계정 및 DynamoDB 테이블에만 유효합니다.

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS Key Management Service 개발자 안내서의 에서 권한 부여 AWS KMS를 참조하세요.

- 자세한 API 내용은 명령 참조ListGrants의 섹션을 참조하세요. AWS CLI

### list-key-policies

다음 코드 예시에서는 list-key-policies을 사용하는 방법을 보여 줍니다.

AWS CLI

KMS 키의 키 정책 이름을 가져오려면

다음 list-key-policies 예시에서는 예시 계정 및 리전의 고객 관리형 키에 대한 키 정책 이름을 가져옵니다. 이 명령을 사용하여 AWS 관리형 키 및 고객 관리형 키의 키 정책 이름을 찾을 수 있습니다.

유효한 키 정책 이름은 default뿐이므로 이 명령은 유용하지 않습니다.

KMS 키를 지정하려면 key-id 파라미터를 사용합니다. 이 예제에서는 키 ID 값을 사용하지만 이 명령ARN에서 키 ID 또는 키를 사용할 수 있습니다.

```
aws kms list-key-policies \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

출력:

```
{
    "PolicyNames": [
    "default"
    ]
}
```

키 정책에 대한 AWS KMS 자세한 내용은 키 AWS 관리 서비스 개발자 안내서의 에서 키 정책 사용을 AWS KMS 참조하세요.

- 자세한 API 내용은 명령 참조ListKeyPolicies의 섹션을 참조하세요. AWS CLI

### list-key-rotations

다음 코드 예시에서는 list-key-rotations을 사용하는 방법을 보여 줍니다.

AWS CLI

완료된 모든 키 재료 교체에 대한 정보를 검색하려면

다음 list-key-rotations 예제에서는 지정된 키에 대해 완료된 모든 KMS 키 구성 요소 교체에 대한 정보를 나열합니다.

```
aws kms list-key-rotations \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

출력:

```
{
    "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
}
```

자세한 내용은 Key Management Service 개발자 안내서의 키 교체를 참조하세요. AWS

- 자세한 API 내용은 명령 참조ListKeyRotations의 섹션을 참조하세요. AWS CLI

### `list-keys`

다음 코드 예시에서는 `list-keys`을 사용하는 방법을 보여 줍니다.

AWS CLI

계정 및 리전에서 KMS 키를 가져오려면

다음 `list-keys` 예제에서는 계정 및 리전의 KMS 키를 가져옵니다. 이 명령은 AWS 관리형 키와 고객 관리형 키를 모두 반환합니다.

```
aws kms list-keys
```

출력:

```
{
```

```
    "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"
        }
    ]
}
```

자세한 내용은 AWS Key Management Service 개발자 안내서의 키 보기를 참조하세요.

- 자세한 API 내용은 명령 참조ListKeys의 섹션을 참조하세요. AWS CLI

### `list-resource-tags`

다음 코드 예시에서는 `list-resource-tags`을 사용하는 방법을 보여 줍니다.

AWS CLI

KMS 키의 태그를 가져오려면

다음 `list-resource-tags` 예제에서는 KMS 키에 대한 태그를 가져옵니다. KMS 키에 리소스 태그를 추가하거나 교체하려면 `tag-resource` 명령을 사용합니다. 출력에 따르면 이 KMS 키에는 두 개의 리소스 태그가 있으며 각 태그에는 키와 값이 있습니다.

KMS 키를 지정하려면 `key-id` 파라미터를 사용합니다. 이 예제에서는 키 ID 값을 사용하지만 이 명령ARN에서 키 ID 또는 키를 사용할 수 있습니다.

```
aws kms list-resource-tags \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

출력:

```
{
    "Tags": [
    {
        "TagKey": "Dept",
        "TagValue": "IT"
    },
    {
        "TagKey": "Purpose",
        "TagValue": "Test"
    }
    ],
    "Truncated": false
}
```

에서 태그를 사용하는 방법에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 키 태그 지정을 AWS KMS참조하세요.

- 자세한 API 내용은 명령 참조ListResourceTags의 섹션을 참조하세요. AWS CLI

### `list-retirable-grants`

다음 코드 예시에서는 `list-retirable-grants`을 사용하는 방법을 보여 줍니다.

AWS CLI

보안 주체가 사용할 수 있는 권한 부여를 보려면

다음 `list-retirable-grants` 예제에서는 `ExampleAdmin` 사용자가 AWS 계정 및 리전의 KMS 키에서 사용 중지할 수 있는 모든 권한을 표시합니다. 이 명령과 같은 명령을 사용하여 계정 보안 주체가 AWS 계정 및 리전의 KMS 키에서 사용 중지할 수 있는 권한을 볼 수 있습니다.

필수 `retiring-principal` 파라미터의 값은 계정, 사용자 또는 역할의 Amazon 리소스 이름(ARN)이어야 합니다.

서비스가 사용 중지 보안 주체일 수 있더라도 이 명령`retiring-principal`에서 값에 대한 서비스를 지정할 수 없습니다. 특정 서비스가 사용 중지 보안 주체인 권한을 찾으려면 `list-grants` 명령을 사용합니다.

출력에 따르면 `ExampleAdmin` 사용자는 계정과 리전의 서로 다른 두 KMS 키에 대한 권한 부여를 중지할 수 있는 권한이 있습니다. 사용 중지된 보안 주체 외에도 계정은 계정의 모든 권한 부여를 사용 중지할 수 있는 권한이 있습니다.

```
aws kms list-retirable-grants \
    --retiring-principal arn:aws:iam::111122223333:user/ExampleAdmin
```

출력:

```
{
    "Grants": [
        {
            "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "GrantId":
 "156b69c63cb154aa21f59929ff19760717be8d9d82b99df53e18b94a15a5e88e",
            "Name": "",
            "CreationDate": 2021-01-14T20:17:36.419000+00:00,
            "GranteePrincipal": "arn:aws:iam::111122223333:user/ExampleUser",
            "RetiringPrincipal": "arn:aws:iam::111122223333:user/ExampleAdmin",
            "IssuingAccount": "arn:aws:iam::111122223333:root",
            "Operations": [
                "Encrypt"
            ],
            "Constraints": {
                "EncryptionContextSubset": {
                    "Department": "IT"
                }
            }
        },
        {
            "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321",
            "GrantId":
 "8c94d1f12f5e69f440bae30eaec9570bb1fb7358824f9ddfa1aa5a0dab1a59b2",
            "Name": "",
            "CreationDate": "2021-02-02T19:49:49.638000+00:00",
            "GranteePrincipal": "arn:aws:iam::111122223333:role/ExampleRole",
            "RetiringPrincipal": "arn:aws:iam::111122223333:user/ExampleAdmin",
            "IssuingAccount": "arn:aws:iam::111122223333:root",
            "Operations": [
                "Decrypt"
            ],
            "Constraints": {
                "EncryptionContextSubset": {
                    "Department": "IT"
                }
```

```
            }
        }
    ],
    "Truncated": false
}
```

자세한 내용은 AWS Key Management Service 개발자 안내서의 에서 권한 부여 AWS KMS를 참조하세요.

- 자세한 API 내용은 명령 참조ListRetirableGrants의 섹션을 참조하세요. AWS CLI

### put-key-policy

다음 코드 예시에서는 put-key-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

키에 대한 KMS 키 정책을 변경하려면

다음 put-key-policy 예시에서는 고객 관리형 키의 키 정책을 변경합니다.

시작하려면 키 정책을 생성하고 로컬 JSON 파일에 저장합니다. 이 예시에서 파일은 key_policy.json입니다. 키 정책을 policy 파라미터의 문자열 값으로 지정할 수도 있습니다.

이 키 정책의 첫 번째 문은 AWS 계정에 IAM 정책을 사용하여 KMS 키에 대한 액세스를 제어할 수 있는 권한을 부여합니다. 두 번째 문은 test-user 사용자에게 KMS 키에서 describe-key 및 list-keys 명령을 실행할 수 있는 권한을 부여합니다.

key_policy.json의 콘텐츠:

```
{
    "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" : "*"
        }
    ]
}
```

KMS 키를 식별하기 위해 이 예제에서는 키 ID를 사용하지만 키 도 사용할 수 있습니다ARN. 키 정책을 지정하기 위해 이 명령은 policy 파라미터를 사용합니다. 정책이 파일에 있음을 나타내기 위해 필수 file:// 접두사를 사용합니다. 이 접두사는 지원되는 모든 운영 체제에서 파일을 식별하는 데 필요합니다. 마지막으로, 이 명령은 값이 default인 policy-name 파라미터를 사용합니다. 정책 이름이 지정되지 않은 경우 기본값은 입니다default. 유일한 유효 값은 default입니다.

```
aws kms put-key-policy \
    --policy-name default \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --policy file://key_policy.json
```

이 명령은 출력을 생성하지 않습니다. 명령이 유효한지 확인하려면 get-key-policy 명령을 사용하세요. 다음 예제 명령은 동일한 키에 대한 KMS 키 정책을 가져옵니다. 값이 text인 output 파라미터는 읽기 쉬운 텍스트 형식을 반환합니다.

```
aws kms get-key-policy \
    --policy-name default \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --output text
```

출력:

```
{
    "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" : "*"
        }
    ]
}
```

자세한 내용은 AWS Key Management Service 개발자 안내서의 키 정책 변경을 참조하세요.

- 자세한 API 내용은 명령 참조PutKeyPolicy의 섹션을 참조하세요. AWS CLI

### `re-encrypt`

다음 코드 예시에서는 `re-encrypt`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 다른 대칭 KMS 키(Linux 및 macOS )로 암호화된 메시지를 다시 암호화하려면

다음 `re-encrypt` 명령 예제는 를 사용하여 데이터를 다시 암호화하는 권장 방법을 보여줍니다 AWS CLI.

file.In `--ciphertext-blob` 파라미터 값을 암호 텍스트에 입력하고 `fileb://` 접두사를 사용합니다. 접두사는 CLI에 바이너리 파일에서 데이터를 읽도록 지시합니다. 파일이 현재 디렉터리에 없는 경우 파일의 전체 경로를 입력합니다. 파일에서 파라미터 값을 읽 AWS CLI는 방법에 대한 자세한 내용은 AWS 명령줄 도구 블로그의 명령줄 인터페이스 사용 설명서 및 로컬 파일 파라미터 모범 사례<https://aws.amazon.com/blogs/developer/best-practices-for-local-file-parameters/>의 파

일 <-https://docs.aws.amazon.com/cli/latest/userguide/cliusage-parameters-file.html>에서 파라미터 AWS CLI로드를 참조하세요. 암호 텍스트를 복호화하는 소스 KMS 키를 지정하세요. 대칭 암호화 KMS 키를 사용하여 복호화할 때는 `--source-key-id` 파라미터가 필요하지 않습니다. AWS KMS 는 암호 텍스트 블롭의 메타데이터에서 데이터를 암호화하는 데 사용된 KMS 키를 가져올 수 있습니다. AWS  하지만 사용하는 KMS 키를 지정하는 것이 항상 모범 사례입니다. 이 방법을 사용하면 의도한 KMS 키를 사용할 수 있으며 신뢰할 수 없는 KMS 키를 사용하여 암호 텍스트를 실수로 복호화하는 것을 방지할 수 있습니다. 데이터를 다시 암호화하는 대상 KMS 키를 지정합니다. `--destination-key-id` 파라미터는 항상 필요합니다. 이 예제에서는 키 를 사용하지ARN만 유효한 키 식별자를 사용할 수 있습니다. 일반 텍스트 출력을 텍스트 값으로 요청합니다. `--query` 파라미터는 출력에서 `Plaintext` 필드 값만 가져오CLI도록 에 지시합니다. `--output` 파라미터는 출력을 텍스트로 반환합니다. 일반 텍스트를 Base64로 디코딩하여 파일에 저장합니다. 다음 예시에서는 `Plaintext` 파라미터 값을 Base64 유틸리티에 파이프(|)로 구분하며 유틸리티가 이를 디코딩합니다. 그런 다음 디코딩된 출력을 `ExamplePlaintext` 파일로 리디렉션(>)합니다.

이 명령을 실행하기 전에 예제 키를 AWS 계정의 유효한 키 식별자IDs로 바꿉니다.

```
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
```

이 명령은 출력을 생성하지 않습니다. `re-encrypt` 명령의 출력은 base64로 디코딩되어 파일에 저장됩니다.

자세한 내용은 AWS Key Management Service API 참조의 ReEncrypt <https://docs.aws.amazon.com/kms/latest/APIReference/API_ReEncrypt.html을 참조하세요.

예제 2: 다른 대칭 KMS 키(Windows 명령 프롬프트)로 암호화된 메시지를 다시 암호화하려면

다음 `re-encrypt` 명령 예시는 `certutil` 유틸리티를 사용하여 일반 텍스트 데이터를 base64로 디코딩한다는 점을 제외하면 이전 예시와 동일합니다. 이 프로시저에는 다음 예시와 같이 두 개의 명령이 필요합니다.

이 명령을 실행하기 전에 예제 키 ID를 AWS 계정의 유효한 키 ID로 바꿉니다.

```
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
```

그런 다음 certutil 유틸리티를 사용하세요.

```
certutil -decode ExamplePlaintextFile.base64 ExamplePlaintextFile
```

출력:

```
Input Length = 18
Output Length = 12
CertUtil: -decode command completed successfully.
```

자세한 내용은 AWS Key Management Service API 참조의 ReEncrypt <https://docs.aws.amazon.com/kms/latest/APIReference/API_ReEncrypt.html을 참조하세요.

- 자세한 API 내용은 명령 참조ReEncrypt의 섹션을 참조하세요. AWS CLI

### retire-grant

다음 코드 예시에서는 retire-grant을 사용하는 방법을 보여 줍니다.

AWS CLI

고객 마스터 키에 대한 권한 부여를 사용 중지하는 방법

다음 retire-grant 예제에서는 KMS 키에서 권한을 삭제합니다.

다음 예시 명령은 grant-id 및 key-id 파라미터를 지정합니다. key-id 파라미터 값은 키ARN의 KMS 키여야 합니다.

```
aws kms retire-grant \
    --grant-id 1234a2345b8a4e350500d432bccf8ecd6506710e1391880c4f7f7140160c9af3 \
    --key-id arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
```

이 명령은 출력을 생성하지 않습니다. 권한 부여가 사용 중지되었는지 확인하려면 list-grants 명령을 사용하세요.

자세한 내용은 AWS Key Management Service 개발자 안내서의 권한 부여 사용 중지 및 취소를 참조하세요.

- 자세한 API 내용은 명령 참조RetireGrant의 섹션을 참조하세요. AWS CLI

### `revoke-grant`

다음 코드 예시에서는 `revoke-grant`을 사용하는 방법을 보여 줍니다.

AWS CLI

고객 마스터 키에 대한 권한 부여를 사용 중지하는 방법

다음 `revoke-grant` 예제에서는 KMS 키에서 권한을 삭제합니다. 다음 예시 명령은 `grant-id` 및 `key-id` 파라미터를 지정합니다. `key-id` 파라미터의 값은 키의 키 ID 또는 키일 수 ARN 있습니다 KMS.

```
aws kms revoke-grant \
    --grant-id 1234a2345b8a4e350500d432bccf8ecd6506710e1391880c4f7f7140160c9af3 \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

이 명령은 출력을 생성하지 않습니다. 권한 부여가 취소되었는지 확인하려면 `list-grants` 명령을 사용하세요.

자세한 내용은 AWS Key Management Service 개발자 안내서의 권한 부여 사용 중지 및 취소를 참조하세요.

- 자세한 API 내용은 명령 참조RevokeGrant의 섹션을 참조하세요. AWS CLI

### `rotate-key-on-demand`

다음 코드 예시에서는 `rotate-key-on-demand`을 사용하는 방법을 보여 줍니다.

AWS CLI

KMS 키의 온디맨드 교체를 수행하려면

다음 `rotate-key-on-demand` 예제에서는 지정된 키에 대한 KMS 키 구성 요소의 교체를 즉시 시작합니다.

```
aws kms rotate-key-on-demand \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

출력:

```
{
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
}
```

자세한 내용은 Key Management Service 개발자 안내서의 온디맨드 키 교체를 수행하는 방법을 참조하세요. AWS

- 자세한 API 내용은 명령 참조RotateKeyOnDemand의 섹션을 참조하세요. AWS CLI

### schedule-key-deletion

다음 코드 예시에서는 schedule-key-deletion을 사용하는 방법을 보여 줍니다.

AWS CLI

고객 관리형 KMS 키 삭제를 예약합니다.

다음 schedule-key-deletion 예제에서는 지정된 고객 관리형 KMS 키를 15일 후에 삭제하도록 예약합니다.

--key-id 파라미터는 KMS 키를 식별합니다. 이 예제에서는 키 ARN 값을 사용하지만 키 ID 또는 KMS 키ARN의 를 사용할 수 있습니다. --pending-window-in-days 파라미터는 7~30일 대기 기간의 길이를 지정합니다. 기본 대기 기간은 30일입니다. 이 예제에서는 15의 값을 지정합니다. 이 값은 명령이 완료된 후 15일 후에 KMS 키를 영구적으로 삭제 AWS 하도록 지시합니다.

```
aws kms schedule-key-deletion \
    --key-id arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab \
    --pending-window-in-days 15
```

응답에는 키 ARN, 키 상태, 대기 기간(PendingWindowInDays) 및 Unix 시간의 삭제 날짜가 포함됩니다. 삭제 날짜를 현지 시간으로 보려면 콘솔을 AWS KMS 사용합니다. KMS PendingDeletion 키 상태의 키는 암호화 작업에 사용할 수 없습니다.

```
{
    "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
}
```

자세한 내용은 AWS Key Management Service 개발자 안내서의 키 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조ScheduleKeyDeletion의 섹션을 참조하세요. AWS CLI

### `sign`

다음 코드 예시에서는 `sign`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 메시지에 대한 디지털 서명을 생성하려면

다음 `sign` 예제에서는 짧은 메시지에 대한 암호화 서명을 생성합니다. 명령 출력에는 `verify` 명령을 사용하여 확인할 수 있는 base-64 인코딩 `Signature` 필드가 포함됩니다.

서명할 메시지와 비대칭 KMS 키가 지원하는 서명 알고리즘을 지정해야 합니다. KMS 키의 서명 알고리즘을 가져오려면 `describe-key` 명령을 사용합니다.

2.0에서 AWS CLI `message` 파라미터 값은 Base64-encoded이어야 합니다. 또는 메시지를 파일에 저장하고 `fileb://` 에 AWS CLI 파일에서 바이너리 데이터를 읽도록 지시하는 접두사를 사용할 수 있습니다.

이 명령을 실행하기 전에 예제 키 ID를 AWS 계정의 유효한 키 ID로 바꿉니다. 키 ID는 KMS 키 사용량이 SIGN_인 비대칭 키를 나타내야 합니다VERIFY.

```
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
```

출력:

```
{
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
```

```
    "Signature": "ABCDEFhpyVYyTxbafE74ccSvEJLJr3zuoV1Hfymz4qv+/
fxmxNLA7SE1SiF8lHw80fKZZ3bJ...",
    "SigningAlgorithm": "RSASSA_PKCS1_V1_5_SHA_256"
}
```

에서 비대칭 KMS 키를 사용하는 방법에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 에서 비대칭 키를 AWS KMS AWS KMS참조하세요.

예제 2: 디지털 서명을 파일에 저장하려면(Linux 및 macOs)

다음 `sign` 예제에서는 로컬 파일에 저장된 짧은 메시지에 대한 암호화 서명을 생성합니다. 또한 명령은 응답에서 `Signature` 속성을 가져오고 Base64-decodes하여 ExampleSignature 파일에 저장합니다. 서명을 확인하는 `verify` 명령에서 서명 파일을 사용할 수 있습니다.

`sign` 명령에는 Base64-encoded 메시지와 비대칭 KMS 키가 지원하는 서명 알고리즘이 필요합니다. KMS 키가 지원하는 서명 알고리즘을 가져오려면 `describe-key` 명령을 사용합니다.

이 명령을 실행하기 전에 예제 키 ID를 AWS 계정의 유효한 키 ID로 바꿉니다. 키 ID는 KMS 키 사용량이 SIGN_인 비대칭 키를 나타내야 합니다VERIFY.

```
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
```

이 명령은 출력을 생성하지 않습니다. 이 예제에서는 출력의 `Signature` 속성을 추출하여 파일에 저장합니다.

에서 비대칭 KMS 키를 사용하는 방법에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 에서 비대칭 키를 AWS KMS AWS KMS참조하세요.

- 자세한 API 내용은 로그인 AWS CLI 명령 참조를 참조하세요.

### `tag-resource`

다음 코드 예시에서는 `tag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

KMS 키에 태그를 추가하려면

다음 `tag-resource` 예제에서는 고객 관리형 KMS 키에 `"Purpose":"Test"` 및 `"Dept":"IT"` 태그를 추가합니다. 이와 같은 태그를 사용하여 KMS 키에 레이블을 지정하고 권한 및 감사를 위한 KMS 키 범주를 생성할 수 있습니다.

KMS 키를 지정하려면 `key-id` 파라미터를 사용합니다. 이 예제에서는 키 ID 값을 사용하지만 이 명령ARN에서 키 ID 또는 키를 사용할 수 있습니다.

```
aws kms tag-resource \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --tags TagKey='Purpose',TagValue='Test' TagKey='Dept',TagValue='IT'
```

이 명령은 출력을 생성하지 않습니다. KMS 키에서 태그를 AWS KMS 보려면 `list-resource-tags` 명령을 사용합니다.

에서 태그를 사용하는 방법에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 키 태그 지정을 AWS KMS참조하세요.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### `untag-resource`

다음 코드 예시에서는 `untag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

KMS 키에서 태그를 삭제하려면

다음 `untag-resource` 예제에서는 고객 관리형 `"Purpose"` 키에서 KMS 키가 있는 태그를 삭제합니다.

KMS 키를 지정하려면 `key-id` 파라미터를 사용합니다. 이 예제에서는 키 ID 값을 사용하지만 이 명령ARN에서 키 ID 또는 키를 사용할 수 있습니다. 이 명령을 실행하기 전에 예제 키 ID를 AWS 계정의 유효한 키 ID로 바꿉니다.

```
aws kms untag-resource \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
```

```
    --tag-key 'Purpose'
```

이 명령은 출력을 생성하지 않습니다. KMS 키에서 태그를 AWS KMS 보려면 `list-resource-tags` 명령을 사용합니다.

에서 태그를 사용하는 방법에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 키 태그 지정을 AWS KMS참조하세요.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-alias

다음 코드 예시에서는 `update-alias`을 사용하는 방법을 보여 줍니다.

AWS CLI

별칭을 다른 KMS 키와 연결하려면

다음 `update-alias` 예제에서는 별칭을 다른 KMS 키`alias/test-key`와 연결합니다.

`--alias-name` 파라미터는 별칭을 지정합니다. 별칭 이름 값은 로 시작해야 합니다`alias/`. `--target-key-id` 파라미터는 별칭과 연결할 KMS 키를 지정합니다. 별칭의 현재 KMS 키를 지정할 필요가 없습니다.

```
aws kms update-alias \
    --alias-name alias/test-key \
    --target-key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

이 명령은 출력을 생성하지 않습니다. 별칭을 찾으려면 `list-aliases` 명령을 사용하세요.

자세한 내용은 AWS Key Management Service 개발자 안내서의 별칭 업데이트를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateAlias의 섹션을 참조하세요. AWS CLI

### update-custom-key-store

다음 코드 예시에서는 `update-custom-key-store`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 사용자 지정 키 스토어의 표시 이름을 편집하려면

다음 update-custom-key-store 예제에서는 사용자 지정 키 스토어의 이름을 변경합니다. 이 예제는 AWS 클라우드HSM 키 스토어 또는 외부 키 스토어에서 작동합니다.

custom-key-store-id 를 사용하여 키 스토어를 식별합니다. new-custom-key-store-name 파라미터를 사용하여 새 표시 이름을 지정합니다.

AWS CloudHSM 키 스토어의 표시 이름을 업데이트하려면 먼저 disconnect-custom-key-store 명령을 사용하여 키 스토어의 연결을 해제해야 합니다. 외부 키 스토어가 연결되거나 연결이 끊긴 상태에서 해당 스토어의 표시 이름을 업데이트할 수 있습니다. 사용자 지정 키 스토어의 연결 상태를 찾으려면 describe-custom-key-store 명령을 사용합니다.

```
aws kms update-custom-key-store \
    --custom-key-store-id cks-1234567890abcdef0 \
    --new-custom-key-store-name ExampleKeyStore
```

이 명령은 데이터를 반환하지 않습니다. 명령이 작동하는지 확인하려면 describe-custom-key-stores 명령을 사용합니다.

AWS CloudHSM 키 스토어 업데이트에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 AWS CloudHSM 키 스토어 설정 편집을 참조하세요.

외부 키 스토어 업데이트에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 외부 키 스토어 속성 편집을 참조하세요.

예제 2: AWS CloudHSM 키 스토어의 kmsuser 암호를 편집하려면

다음 update-custom-key-store 예제에서는 지정된 키 스토어와 연결된 클라우드HSM 클러스터kmsuser에서 kmsuser 암호 값을 의 현재 암호로 업데이트합니다. 이 명령은 클러스터의 kmsuser 암호를 변경하지 않습니다. 현재 암호를 알려줍니다 AWS KMS. 에 현재 kmsuser 암호가 KMS 없는 경우 AWS 클라우드HSM 키 스토어에 연결할 수 없습니다.

NOTE: AWS CloudHSM 키 스토어를 업데이트하기 전에 연결을 해제해야 합니다. disconnect-custom-key-store 명령을 사용합니다. 명령이 완료되면 AWS CloudHSM 키 스토어를 다시 연결할 수 있습니다. connect-custom-key-store 명령을 사용합니다.

```
aws kms update-custom-key-store \
    --custom-key-store-id cks-1234567890abcdef0 \
    --key-store-password ExamplePassword
```

이 명령은 출력을 반환하지 않습니다. 변경 사항이 유효한지 확인하려면 describe-custom-key-stores 명령을 사용합니다.

AWS CloudHSM 키 스토어 업데이트에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 AWS CloudHSM 키 스토어 설정 편집을 참조하세요.

예제 3: AWS CloudHSM 키 스토어의 AWS CloudHSM 클러스터 편집

다음 예제에서는 AWS CloudHSM 키 스토어와 연결된 AWS CloudHSM 클러스터를 동일한 클러스터의 다른 백업과 같은 관련 클러스터로 변경합니다.

NOTE: AWS CloudHSM 키 스토어를 업데이트하기 전에 연결을 해제해야 합니다. `disconnect-custom-key-store` 명령을 사용합니다. 명령이 완료되면 AWS CloudHSM 키 스토어를 다시 연결할 수 있습니다. `connect-custom-key-store` 명령을 사용합니다.

```
aws kms update-custom-key-store \
    --custom-key-store-id cks-1234567890abcdef0 \
    --cloud-hsm-cluster-id cluster-1a23b4cdefg
```

이 명령은 출력을 반환하지 않습니다. 변경 사항이 유효한지 확인하려면 `describe-custom-key-stores` 명령을 사용합니다.

AWS CloudHSM 키 스토어 업데이트에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 AWS CloudHSM 키 스토어 설정 편집을 참조하세요.

예제 4: 외부 키 스토어의 프록시 인증 자격 증명을 편집하려면

다음 예제에서는 외부 키 스토어의 프록시 인증 자격 증명을 업데이트합니다. 값 중 하나만 변경access-key-id하더라도 `raw-secret-access-key` 및 를 모두 지정해야 합니다. 이 기능을 사용하여 잘못된 자격 증명을 수정하거나 외부 키 스토어 프록시가 자격 증명을 교체할 때 자격 증명을 변경할 수 있습니다.

외부 키 스토어에서 에 대한 AWS KMS 프록시 인증 자격 증명을 설정합니다. 그런 다음 이 명령을 사용하여 에 자격 증명을 제공합니다 AWS KMS. AWS KMS 는 이 자격 증명을 사용하여 외부 키 스토어 프록시에 대한 요청에 서명합니다.

외부 키 스토어가 연결되거나 연결이 끊긴 상태에서 프록시 인증 보안 인증을 업데이트할 수 있습니다. 사용자 지정 키 스토어의 연결 상태를 찾으려면 `describe-custom-key-store` 명령을 사용합니다.

```
aws kms update-custom-key-store \
    --custom-key-store-id cks-1234567890abcdef0 \
    --xks-proxy-authentication-credential "AccessKeyId=ABCDE12345670EXAMPLE,
 RawSecretAccessKey=DXjSUawnel2fr6SKC7G25CNxTyWKE5PF9XX6H/u9pSo="
```

이 명령은 출력을 반환하지 않습니다. 변경 사항이 유효한지 확인하려면 `describe-custom-key-stores` 명령을 사용합니다.

외부 키 스토어 업데이트에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 외부 키 스토어 속성 편집을 참조하세요.

예제 5: 외부 키 스토어의 프록시 연결을 편집하려면

다음 예제에서는 외부 키 스토어 프록시 연결 옵션을 퍼블릭 엔드포인트 연결에서 VPC 엔드포인트 서비스 연결로 변경합니다. `xks-proxy-connectivity` 값을 변경하는 것 외에도 VPC 엔드포인트 서비스와 연결된 프라이빗 DNS 이름을 반영하도록 `xks-proxy-uri-endpoint` 값을 변경해야 합니다. 값을 추가해야 합니다`xks-proxy-vpc-endpoint-service-name`.

NOTE: 외부 스토어의 프록시 연결을 업데이트하기 전에 연결을 해제해야 합니다. `disconnect-custom-key-store` 명령을 사용합니다. 명령이 완료되면 `connect-custom-key-store` 명령을 사용하여 외부 키 스토어를 다시 연결할 수 있습니다.

```
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"
```

이 명령은 출력을 반환하지 않습니다. 변경 사항이 유효한지 확인하려면 `describe-custom-key-stores` 명령을 사용합니다.

외부 키 스토어 업데이트에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 외부 키 스토어 속성 편집을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateCustomKeyStore의 섹션을 참조하세요. AWS CLI

### update-key-description

다음 코드 예시에서는 `update-key-description`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 고객 관리형 KMS 키에 설명을 추가하거나 변경하려면

다음 `update-key-description` 예제에서는 고객 관리형 KMS 키에 설명을 추가합니다. 동일한 명령을 사용하여 기존 설명을 변경할 수 있습니다.

--key-id 파라미터는 명령의 KMS 키를 식별합니다. 이 예제에서는 키 ARN 값을 사용하지만 키 ID 또는 KMS 키ARN의 키를 사용할 수 있습니다. --description 파라미터는 새 설명을 지정합니다. 이 파라미터의 값은 KMS 키에 대한 현재 설명이 있는 경우 이를 대체합니다.

```
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"
```

이 명령은 출력을 생성하지 않습니다. KMS 키에 대한 설명을 보려면 describe-key 명령을 사용합니다.

자세한 내용은 키 관리 서비스 참조UpdateKeyDescription의 섹션을 참조하세요. AWS API

예제 2: 고객 관리형 KMS 키에 대한 설명을 삭제하려면

다음 update-key-description 예제에서는 고객 관리형 KMS 키에 대한 설명을 삭제합니다.

--key-id 파라미터는 명령의 KMS 키를 식별합니다. 이 예제에서는 키 ID 값을 사용하지만 키 ID 또는 KMS 키ARN의 키를 사용할 수 있습니다. 빈 문자열 값('')이 있는 --description 파라미터는 기존 설명을 삭제합니다.

```
aws kms update-key-description \
    --key-id 0987dcba-09fe-87dc-65ba-ab0987654321 \
    --description ''
```

이 명령은 출력을 생성하지 않습니다. KMS 키에 대한 설명을 보려면 describe-key 명령을 사용합니다.

자세한 내용은 키 관리 서비스 참조UpdateKeyDescription의 섹션을 참조하세요. AWS API

- 자세한 API 내용은 명령 참조UpdateKeyDescription의 섹션을 참조하세요. AWS CLI

### verify

다음 코드 예시에서는 verify을 사용하는 방법을 보여 줍니다.

AWS CLI

디지털 서명을 확인하려면

다음 verify 예제에서는 짧은 Base64-encoded 메시지에 대한 암호화 서명을 확인합니다. 키 ID, 메시지, 메시지 유형 및 서명 알고리즘은 메시지에 서명하는 데 사용된 것과 동일해야 합니다. 지정한 서명은 base64로 인코딩할 수 없습니다. sign 명령이 반환하는 서명을 디코딩하는 데 도움이 필요하면 sign 명령 예제를 참조하세요.

명령의 출력에는 서명이 확인되었음을 나타내는 부울 SignatureValid 필드가 포함됩니다. 서명 검증에 실패하면 verify 명령도 실패합니다.

이 명령을 실행하기 전에 예제 키 ID를 AWS 계정의 유효한 키 ID로 바꿉니다.

```
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
```

출력:

```
{
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "SignatureValid": true,
    "SigningAlgorithm": "RSASSA_PKCS1_V1_5_SHA_256"
}
```

에서 비대칭 KMS 키 사용에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 비대칭 키 사용을 AWS KMS참조하세요.

- 자세한 API 내용은 명령 참조의 확인을 참조하세요. AWS CLI

## 를 사용한 Lake Formation 예제 AWS CLI

다음 코드 예제에서는 Lake Formation과 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### add-lf-tags-to-resource

다음 코드 예시에서는 add-lf-tags-to-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

기존 리소스에 하나 이상의 LF 태그를 연결하려면

다음 add-lf-tags-to-resource 예제에서는 지정된 LF 태그를 테이블 리소스에 연결합니다.

```
aws lakeformation add-lf-tags-to-resource \
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
    "CatalogId": "123456789111",
    "Resource": {
        "Table": {
            "CatalogId": "123456789111",
            "DatabaseName": "tpc",
            "Name": "dl_tpc_promotion"
        }
    },
    "LFTags": [{
        "CatalogId": "123456789111",
        "TagKey": "usergroup",
        "TagValues": [
            "analyst"
        ]
    }]
}
```

출력:

```
{
    "Failures": []
}
```

자세한 내용은 AWS Lake Formation 개발자 안내서의 데이터 카탈로그 리소스에 LF 태그 할당을 참조하세요.

- 자세한 API 내용은 명령 참조AddLfTagsToResource의 섹션을 참조하세요. AWS CLI

### batch-grant-permissions

다음 코드 예시에서는 batch-grant-permissions을 사용하는 방법을 보여 줍니다.

AWS CLI

보안 주체에게 리소스에 대한 권한을 대량 부여하려면

다음 batch-grant-permissions 예제 대량은 보안 주체에게 지정된 리소스에 대한 액세스 권한을 부여합니다.

```
aws lakeformation batch-grant-permissions \
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
    "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": []
        }
    ]
}
```

출력:

```
{
    "Failures": []
}
```

자세한 내용은 AWS Lake Formation 개발자 안내서의 데이터 카탈로그 리소스에 대한 권한 부여 및 취소를 참조하세요.

- 자세한 API 내용은 명령 참조BatchGrantPermissions의 섹션을 참조하세요. AWS CLI

### `batch-revoke-permissions`

다음 코드 예시에서는 `batch-revoke-permissions`을 사용하는 방법을 보여 줍니다.

AWS CLI

보안 주체의 리소스에 대한 권한 대량 취소

다음 batch-revoke-permissions 예제 대량은 보안 주체로부터 지정된 리소스에 대한 액세스를 취소합니다.

```
aws lakeformation batch-revoke-permissions \
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
    "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"
            ]
        }
    ]
}
```

출력:

```
{
    "Failures": []
}
```

자세한 내용은 AWS Lake Formation 개발자 안내서의 데이터 카탈로그 리소스에 대한 권한 부여 및 취소를 참조하세요.

- 자세한 API 내용은 명령 참조BatchRevokePermissions의 섹션을 참조하세요. AWS CLI

### `cancel-transaction`

다음 코드 예시에서는 `cancel-transaction`을 사용하는 방법을 보여 줍니다.

AWS CLI

트랜잭션을 취소하려면

다음 `cancel-transaction` 예제에서는 트랜잭션을 취소합니다.

```
aws lakeformation cancel-transaction \
    --transaction-id='b014d972ca8347b89825e33c5774aec4'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Lake Formation 개발자 안내서의 트랜잭션 내에서 데이터 레이크에서 읽기 및 쓰기를 참조하세요.

- 자세한 API 내용은 명령 참조CancelTransaction의 섹션을 참조하세요. AWS CLI

### commit-transaction

다음 코드 예시에서는 commit-transaction을 사용하는 방법을 보여 줍니다.

AWS CLI

트랜잭션 커밋

다음 commit-transaction 예제에서는 트랜잭션을 커밋합니다.

```
aws lakeformation commit-transaction \
    --transaction-id='b014d972ca8347b89825e33c5774aec4'
```

출력:

```
{
    "TransactionStatus": "committed"
}
```

자세한 내용은 AWS Lake Formation 개발자 안내서의 트랜잭션 내에서 데이터 레이크에서 읽기 및 쓰기를 참조하세요.

- 자세한 API 내용은 명령 참조CommitTransaction의 섹션을 참조하세요. AWS CLI

### create-data-cells-filter

다음 코드 예시에서는 create-data-cells-filter을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 데이터 셀 필터를 생성하려면

다음 create-data-cells-filter 예제에서는 행 조건에 따라 특정 열에 대한 액세스 권한을 부여할 수 있도록 데이터 셀 필터를 생성합니다.

```
aws lakeformation create-data-cells-filter \
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
```

```
    "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"
    }
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Lake Formation 개발자 안내서의 Lake Formation의 데이터 필터링 및 셀 수준 보안을 참조하세요. AWS

예제 2: 열 필터 생성

다음 `create-data-cells-filter` 예제에서는 데이터 필터를 생성하여 특정 열에 대한 액세스 권한을 부여합니다.

```
aws lakeformation create-data-cells-filter \
    --cli-input-json file://input.json
```

`input.json`의 콘텐츠:

```
{
    "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"
    }
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Lake Formation 개발자 안내서의 Lake Formation의 데이터 필터링 및 셀 수준 보안을 참조하세요. AWS

예제 3: 제외 열로 데이터 필터를 생성하려면

다음 `create-data-cells-filter` 예제에서는 언급된 열을 제외한 모든 열에 대한 액세스 권한을 부여할 수 있는 데이터 필터를 생성합니다.

```
aws lakeformation create-data-cells-filter \
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
    "TableData": {
        "ColumnWildcard": {
            "ExcludedColumnNames": ["p_channel_details", "p_start_date_sk"]
        },
        "DatabaseName": "tpc",
        "Name": "developer_promotion_excludecolumn",
        "RowFilter": {
            "AllRowsWildcard": {}
        },
        "TableCatalogId": "123456789111",
        "TableName": "dl_tpc_promotion"
    }
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Lake Formation 개발자 안내서의 Lake Formation의 데이터 필터링 및 셀 수준 보안을 참조하세요. AWS

- 자세한 API 내용은 명령 참조CreateDataCellsFilter의 섹션을 참조하세요. AWS CLI

### create-lf-tag

다음 코드 예시에서는 `create-lf-tag`을 사용하는 방법을 보여 줍니다.

AWS CLI

LF 태그를 생성하려면

다음 create-lf-tag 예제에서는 지정된 이름과 값을 사용하여 LF 태그를 생성합니다.

```
aws lakeformation create-lf-tag \
    --catalog-id '123456789111' \
    --tag-key 'usergroup' \
    --tag-values '["developer","analyst","campaign"]'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Lake Formation 개발자 안내서의 메타데이터 액세스 제어를 위한 LF 태그 관리를 참조하세요.

- 자세한 API 내용은 명령 참조CreateLfTag의 섹션을 참조하세요. AWS CLI

### delete-data-cells-filter

다음 코드 예시에서는 delete-data-cells-filter을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 셀 필터를 삭제하려면

다음 delete-data-cells-filter 예제에서는 지정된 데이터 셀 필터를 삭제합니다.

```
aws lakeformation delete-data-cells-filter \
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
    "TableCatalogId": "123456789111",
    "DatabaseName": "tpc",
    "TableName": "dl_tpc_promotion",
    "Name": "developer_promotion"
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Lake Formation 개발자 안내서의 Lake Formation의 데이터 필터링 및 셀 수준 보안을 참조하세요. AWS

- 자세한 API 내용은 명령 참조DeleteDataCellsFilter의 섹션을 참조하세요. AWS CLI

### delete-lf-tag

다음 코드 예시에서는 delete-lf-tag을 사용하는 방법을 보여 줍니다.

AWS CLI

LF 태그 정의를 삭제하려면

다음 delete-lf-tag 예제에서는 LF 태그 정의를 삭제합니다.

```
aws lakeformation delete-lf-tag \
    --catalog-id '123456789111' \
    --tag-key 'usergroup'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Lake Formation 개발자 안내서의 메타데이터 액세스 제어를 위한 LF 태그 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteLfTag의 섹션을 참조하세요. AWS CLI

### delete-objects-on-cancel

다음 코드 예시에서는 delete-objects-on-cancel을 사용하는 방법을 보여 줍니다.

AWS CLI

트랜잭션이 취소될 때 객체를 삭제하려면

다음 delete-objects-on-cancel 예제에서는 트랜잭션이 취소될 때 나열된 s3 객체를 삭제합니다.

```
aws lakeformation delete-objects-on-cancel \
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
    "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"
    }]
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Lake Formation 개발자 안내서의 트랜잭션 내 데이터 레이크에서 읽기 및 쓰기를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteObjectsOnCancel의 섹션을 참조하세요. AWS CLI

### deregister-resource

다음 코드 예시에서는 deregister-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 레이크 스토리지 등록을 취소하려면

다음 deregister-resource 예제에서는 Lake Formation에서 관리하는 대로 리소스를 등록 취소합니다.

```
aws lakeformation deregister-resource \
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
    "ResourceArn": "arn:aws:s3:::lf-emr-athena-result-123"
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Lake Formation 개발자 안내서의 데이터 레이크에 Amazon S3 위치 추가를 참조하세요.

- 자세한 API 내용은 명령 참조DeregisterResource의 섹션을 참조하세요. AWS CLI

### describe-transaction

다음 코드 예시에서는 describe-transaction을 사용하는 방법을 보여 줍니다.

AWS CLI

트랜잭션 세부 정보를 검색하려면

다음 describe-transaction 예제에서는 단일 트랜잭션의 세부 정보를 반환합니다.

```
aws lakeformation describe-transaction \
    --transaction-id='8cb4b1a7cc8d486fbaca9a64e7d9f5ce'
```

출력:

```
{
    "TransactionDescription": {
        "TransactionId": "12345972ca8347b89825e33c5774aec4",
        "TransactionStatus": "committed",
        "TransactionStartTime": "2022-08-10T14:29:04.046000+00:00",
        "TransactionEndTime": "2022-08-10T14:29:09.681000+00:00"
    }
}
```

자세한 내용은 AWS Lake Formation 개발자 안내서의 트랜잭션 내에서 데이터 레이크에서 읽기 및 쓰기를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeTransaction의 섹션을 참조하세요. AWS CLI

### extend-transaction

다음 코드 예시에서는 extend-transaction을 사용하는 방법을 보여 줍니다.

AWS CLI

트랜잭션을 확장하려면

다음 extend-transaction 예제는 트랜잭션을 확장합니다.

```
aws lakeformation extend-transaction \
```

```
    --transaction-id='8cb4b1a7cc8d486fbaca9a64e7d9f5ce'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Lake Formation 개발자 안내서의 트랜잭션 내 데이터 레이크에서 읽기 및 쓰기를 참조하세요.

- 자세한 API 내용은 명령 참조ExtendTransaction의 섹션을 참조하세요. AWS CLI

### get-data-lake-settings

다음 코드 예시에서는 get-data-lake-settings을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS Lake Formation 관리형 데이터 레이크 설정을 검색하려면

다음 get-data-lake-settings 예제에서는 데이터 레이크 관리자 및 기타 데이터 레이크 설정 목록을 검색합니다.

```
aws lakeformation get-data-lake-settings \
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
    "CatalogId": "123456789111"
}
```

출력:

```
{
    "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"
        ]
    }
}
```

자세한 내용은 AWS Lake Formation 개발자 안내서의 데이터 레이크에 대한 기본 보안 설정 변경을 참조하세요.

- 자세한 API 내용은 명령 참조GetDataLakeSettings의 섹션을 참조하세요. AWS CLI

### get-effective-permissions-for-path

다음 코드 예시에서는 get-effective-permissions-for-path을 사용하는 방법을 보여 줍니다.

AWS CLI

특정 경로에 있는 리소스에 대한 권한을 검색하려면

다음 get-effective-permissions-for-path 예제에서는 Amazon S3의 경로에 있는 지정된 테이블 또는 데이터베이스 리소스에 대한 Lake Formation 권한을 반환합니다.

```
aws lakeformation get-effective-permissions-for-path \
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
    "CatalogId": "123456789111",
    "ResourceArn": "arn:aws:s3:::lf-data-lake-123456789111"
}
```

출력:

```
{
    "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":
 "E5SlJDSTZleUp6SWpvaU9UQTNORE0zTXpFeE5Ua3pJbjE5TENKbGVIQnBjbUYwYVc5dUlqcDdJbk5sWTI5dVpITWlP
}
```

자세한 내용은 Lake Formation 개발자 안내서의 Lake Formation 권한 관리를 참조하세요. AWS

- 자세한 API 내용은 명령 참조GetEffectivePermissionsForPath의 섹션을 참조하세요. AWS CLI

### `get-lf-tag`

다음 코드 예시에서는 `get-lf-tag`을 사용하는 방법을 보여 줍니다.

AWS CLI

LF 태그 정의를 검색하려면

다음 `get-lf-tag` 예제에서는 LF 태그 정의를 검색합니다.

```
aws lakeformation get-lf-tag \
    --catalog-id '123456789111' \
    --tag-key 'usergroup'
```

출력:

```
{
    "CatalogId": "123456789111",
    "TagKey": "usergroup",
    "TagValues": [
        "analyst",
        "campaign",
        "developer"
    ]
}
```

자세한 내용은 AWS Lake Formation 개발자 안내서의 메타데이터 액세스 제어를 위한 LF 태그 관리를 참조하세요.

- 자세한 API 내용은 명령 참조GetLfTag의 섹션을 참조하세요. AWS CLI

### get-query-state

다음 코드 예시에서는 get-query-state을 사용하는 방법을 보여 줍니다.

AWS CLI

제출된 쿼리의 상태를 검색하려면

다음 get-query-state 예제에서는 이전에 제출한 쿼리의 상태를 반환합니다.

```
aws lakeformation get-query-state \
    --query-id='1234273f-4a62-4cda-8d98-69615ee8be9b'
```

출력:

```
{
    "State": "FINISHED"
}
```

자세한 내용은 AWS Lake Formation 개발자 안내서의 트랜잭션 데이터 작업을 참조하세요.

- 자세한 API 내용은 명령 참조GetQueryState의 섹션을 참조하세요. AWS CLI

### get-query-statistics

다음 코드 예시에서는 get-query-statistics을 사용하는 방법을 보여 줍니다.

AWS CLI

쿼리 통계를 검색하려면

다음 get-query-statistics 예제에서는 쿼리의 계획 및 실행에 대한 통계를 검색합니다.

```
aws lakeformation get-query-statistics \
    --query-id='1234273f-4a62-4cda-8d98-69615ee8be9b'
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS Lake Formation 개발자 안내서의 트랜잭션 데이터 작업을 참조하세요.

- 자세한 API 내용은 명령 참조GetQueryStatistics의 섹션을 참조하세요. AWS CLI

### get-resource-lf-tags

다음 코드 예시에서는 get-resource-lf-tags을 사용하는 방법을 보여 줍니다.

AWS CLI

LF 태그를 나열하려면

다음 list-lf-tags 예제에서는 요청자가 볼 수 있는 권한이 있는 LF 태그 목록을 반환합니다.

```
aws lakeformation list-lf-tags \
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
    "CatalogId": "123456789111",
    "ResourceShareType": "ALL",
    "MaxResults": 2
}
```

출력:

```
{
"LFTags": [{
        "CatalogId": "123456789111",
        "TagKey": "category",
```

```
        "TagValues": [
            "private",
            "public"
        ]
    },
    {
        "CatalogId": "123456789111",
        "TagKey": "group",
        "TagValues": [
            "analyst",
            "campaign",
            "developer"
        ]
    }],
    "NextToken": "kIiwiZXhwaXJhdGlvbiI6eyJzZWNvbmRzIjoxNjYwMDY4dCI6ZmFsc2V9"
}
```

자세한 내용은 AWS Lake Formation 개발자 안내서의 메타데이터 액세스 제어를 위한 LF 태그 관리를 참조하세요.

- 자세한 API 내용은 명령 참조GetResourceLfTags의 섹션을 참조하세요. AWS CLI

### `get-table-objects`

다음 코드 예시에서는 `get-table-objects`을 사용하는 방법을 보여 줍니다.

AWS CLI

관리 테이블의 객체를 나열하려면

다음 `get-table-objects` 예제에서는 지정된 관리 테이블을 구성하는 Amazon S3 객체 세트를 반환합니다.

```
aws lakeformation get-table-objects \
    --cli-input-json file://input.json
```

`input.json`의 콘텐츠:

```
{
    "CatalogId": "012345678901",
    "DatabaseName": "tpc",
    "TableName": "dl_tpc_household_demographics_gov",
```

```
    "QueryAsOfTime": "2022-08-10T15:00:00"
}
```

출력:

```
{
    "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
        }]
    }]
}
```

자세한 내용은 AWS Lake Formation 개발자 안내서의 트랜잭션 내에서 데이터 레이크에서 읽기 및 쓰기를 참조하세요.

- 자세한 API 내용은 명령 참조GetTableObjects의 섹션을 참조하세요. AWS CLI

### get-work-unit-results

다음 코드 예시에서는 `get-work-unit-results`을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 쿼리의 작업 단위를 검색하려면

다음 `get-work-unit-results` 예제에서는 쿼리에서 가져온 작업 단위를 반환합니다.

```
aws lakeformation get-work-units \
    --query-id='1234273f-4a62-4cda-8d98-69615ee8be9b' \
    --work-unit-id '0' \
    --work-unit-token 'B2fMSdmQXe9umX8Ux8XCo4=' outfile
```

출력:

```
outfile with Blob content.
```

자세한 내용은 AWS Lake Formation 개발자 안내서의 트랜잭션 데이터 작업을 참조하세요.

- 자세한 API 내용은 명령 참조GetWorkUnitResults의 섹션을 참조하세요. AWS CLI

### get-work-units

다음 코드 예시에서는 get-work-units을 사용하는 방법을 보여 줍니다.

AWS CLI

작업 단위를 검색하려면

다음 get-work-units 예제에서는 StartQueryPlanning 작업에서 생성된 작업 단위를 검색합니다.

```
aws lakeformation get-work-units \
    --query-id='1234273f-4a62-4cda-8d98-69615ee8be9b'
```

출력:

```
{
    "WorkUnitRanges": [{
        "WorkUnitIdMax": 0,
        "WorkUnitIdMin": 0,
        "WorkUnitToken":
 "1234eMAk4kLO4umqEL4Z5WuxL04AXwABABVhd3MtY3J5cHRvLXB1YmxpYy1rZXkAREEwYm9QbkhINmFYTWphbmMxZW
+f88jzGrYq22gE6jkQlpOB
+Oet2eqNUmFudAAAAfjB8BgkqhkiG9w0BBwagbzBtAgEAMGgGCSqGSIb3DQEHATAeBglghkgBZQMEAS4wEQQMCOEWRda
wAAAAEAAAAAAAAAAAAAAAEAAACX3/w5h75QAPomfKH+cyEKYU1yccUmBl
+VSojiGOtdsUk7vcjYXUUboYm3dvqRqX2s4gROMOn
+Ij8R0/8jYmnHkpvyAFNVRPyETyIKg7k5Z9+5I1c2d3446Jw/moWGGxjH8AEG9h27ytmOhozxDOEi/
F2ZoXz6wlGDfGUo/2WxCkYOhTyNaw6TM
+7drTM7yrW4iNVLUM0LX0xnFjIAhLhooWJek6vjQZUAZzBlAjBH8okRtYP8R7AY2Wls/
hqFBhG0V4l42AC0LxsuZbMQrE2SzWZUZ0E9Uew7/n0cyX4CMQDR79INyv4ysMByW9kKGGKyba+cCNklExMR
+btBQBmMuB2fMSdmQXe9umX8Ux8XCo4="
    }],
    "QueryId": "1234273f-4a62-4cda-8d98-69615ee8be9b"
}
```

자세한 내용은 AWS Lake Formation 개발자 안내서의 트랜잭션 데이터 작업을 참조하세요.

- 자세한 API 내용은 명령 참조GetWorkUnits의 섹션을 참조하세요. AWS CLI

### grant-permissions

다음 코드 예시에서는 grant-permissions을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: LF 태그를 사용하여 리소스에 대한 권한을 보안 주체에게 부여하려면

다음 grant-permissions 예제에서는 LF 태그 정책과 일치하는 데이터베이스 리소스의 보안 주체에 ALL 권한을 부여합니다.

```
aws lakeformation grant-permissions \
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
    "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"
    ]
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Lake Formation 개발자 안내서의 데이터 카탈로그 리소스에 대한 권한 부여 및 취소를 참조하세요.

예제 2: 보안 주체에 열 수준 권한을 부여하려면

다음 grant-permissions 예제에서는 보안 주체에게 특정 열을 선택할 수 있는 권한을 부여합니다.

```
aws lakeformation grant-permissions \
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
    "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": []
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Lake Formation 개발자 안내서의 데이터 카탈로그 리소스에 대한 권한 부여 및 취소를 참조하세요.

예제 3: 보안 주체에 테이블 권한을 부여하려면

다음 grant-permissions 예제에서는 지정된 데이터베이스의 모든 테이블에 대한 선택 권한을 보안 주체에게 부여합니다.

```
aws lakeformation grant-permissions \
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
    "CatalogId": "123456789111",
    "Principal": {
        "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-developer"
    },
    "Resource": {
        "Table": {
            "CatalogId": "123456789111",
            "DatabaseName": "tpc",
            "TableWildcard": {}
        }
    },
    "Permissions": [
        "SELECT"
    ],
    "PermissionsWithGrantOption": []
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Lake Formation 개발자 안내서의 데이터 카탈로그 리소스에 대한 권한 부여 및 취소를 참조하세요.

예제 4: 보안 주체에게 LF 태그에 대한 권한을 부여하려면

다음 grant-permissions 예제에서는 보안 주체에게 LF 태그에 대한 연결 권한을 부여합니다.

```
aws lakeformation grant-permissions \
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
    "CatalogId": "123456789111",
    "Principal": {
        "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-developer"
```

```
    },
    "Resource": {
        "LFTag": {
            "CatalogId": "123456789111",
            "TagKey": "category",
            "TagValues": [
                "private", "public"
            ]
        }

    },
    "Permissions": [
        "ASSOCIATE"
    ],
    "PermissionsWithGrantOption": []
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Lake Formation 개발자 안내서의 데이터 카탈로그 리소스에 대한 권한 부여 및 취소를 참조하세요.

예제 5: 보안 주체에게 데이터 위치에 대한 권한을 부여하려면

다음 `grant-permissions` 예제에서는 보안 주체에게 데이터 위치에 대한 권한을 부여합니다.

```
aws lakeformation grant-permissions \
    --cli-input-json file://input.json
```

`input.json`의 콘텐츠:

```
{
    "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": []
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Lake Formation 개발자 안내서의 데이터 카탈로그 리소스에 대한 권한 부여 및 취소를 참조하세요.

- 자세한 API 내용은 명령 참조GrantPermissions의 섹션을 참조하세요. AWS CLI

### `list-data-cells-filter`

다음 코드 예시에서는 `list-data-cells-filter`을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 셀 필터를 나열하려면

다음 `list-data-cells-filter` 예제에서는 지정된 테이블에 대한 데이터 셀 필터를 나열합니다.

```
aws lakeformation list-data-cells-filter \
    --cli-input-json file://input.json
```

`input.json`의 콘텐츠:

```
{
    "MaxResults": 2,
    "Table": {
        "CatalogId": "123456789111",
        "DatabaseName": "tpc",
        "Name": "dl_tpc_promotion"
    }
}
```

출력:

```
{
    "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"
}
```

자세한 내용은 Lake Formation 개발자 안내서의 Lake Formation의 데이터 필터링 및 셀 수준 보안을 참조하세요. AWS

- 자세한 API 내용은 명령 참조ListDataCellsFilter의 섹션을 참조하세요. AWS CLI

### `list-permissions`

다음 코드 예시에서는 `list-permissions`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 리소스에 대한 보안 주체 권한 목록을 검색하려면

다음 `list-permissions` 예제에서는 데이터베이스 리소스에 대한 보안 주체 권한 목록을 반환합니다.

```
aws lakeformation list-permissions \
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
    "CatalogId": "123456789111",
    "ResourceType": "DATABASE",
    "MaxResults": 2
}
```

출력:

```
{
    "PrincipalResourcePermissions": [{
        "Principal": {
            "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
campaign-manager"
        },
        "Resource": {
            "Database": {
                "CatalogId": "123456789111",
                "Name": "tpc"
            }
        },
        "Permissions": [
            "DESCRIBE"
        ],
        "PermissionsWithGrantOption": []
    }],
    "NextToken":
 "E5SlJDSTZleUp6SWpvaU9UQTNORE0zTXpFeE5Ua3pJbjE5TENKbGVIQnBjbUYwYVc5dUlqcDdJbk5sWTI5dVpITWlP
}
```

자세한 내용은 Lake Formation 개발자 안내서의 Lake Formation 권한 관리를 참조하세요. AWS

예제 2: 데이터 필터를 사용하여 테이블의 보안 주체 권한 목록을 검색하려면

다음 `list-permissions` 예제에서는 보안 주체에게 부여된 관련 데이터 필터와 함께 테이블의 권한을 나열합니다.

```
aws lakeformation list-permissions \
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
    "CatalogId": "123456789111",
    "Resource": {
        "Table": {
            "CatalogId": "123456789111",
            "DatabaseName": "tpc",
            "Name": "dl_tpc_customer"
        }
    },
    "IncludeRelated": "TRUE",
    "MaxResults": 10
}
```

출력:

```
{
    "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"
}
```

자세한 내용은 Lake Formation 개발자 안내서의 Lake Formation 권한 관리를 참조하세요. AWS

예제 3: LF 태그에 대한 보안 주체 권한 목록을 검색하려면

다음 `list-permissions` 예제에서는 보안 주체에게 부여된 LF 태그에 대한 권한을 나열합니다.

```
aws lakeformation list-permissions \
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
    "CatalogId": "123456789111",
    "Resource": {
        "LFTag": {
            "CatalogId": "123456789111",
            "TagKey": "category",
            "TagValues": [
                "private"
            ]
        }
    },
    "MaxResults": 10
}
```

출력:

```
{
    "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": "EJwY21GMGFXOXVJanA3SW5Ocm1pc3Npb25zIjpmYWxzZX0="
}
```

자세한 내용은 Lake Formation 개발자 안내서의 Lake Formation 권한 관리를 참조하세요. AWS

- 자세한 API 내용은 명령 참조ListPermissions의 섹션을 참조하세요. AWS CLI

### `list-resources`

다음 코드 예시에서는 `list-resources`을 사용하는 방법을 보여 줍니다.

AWS CLI

Lake Formation에서 관리하는 리소스를 나열하려면

다음 `list-resources` 예제에서는 Lake Formation에서 관리하는 조건과 일치하는 리소스를 나열합니다.

```
aws lakeformation list-resources \
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
    "FilterConditionList": [{
        "Field": "ROLE_ARN",
        "ComparisonOperator": "CONTAINS",
        "StringValueList": [
            "123456789111"
        ]
    }],
    "MaxResults": 10
}
```

출력:

```
{
    "ResourceInfoList": [{
            "ResourceArn": "arn:aws:s3:::lf-data-lake-123456789111",
            "RoleArn": "arn:aws:iam::123456789111:role/LF-GlueServiceRole",
            "LastModified": "2022-07-21T02:12:46.669000+00:00"
        },
        {
            "ResourceArn": "arn:aws:s3:::lf-emr-test-123456789111",
            "RoleArn": "arn:aws:iam::123456789111:role/EMRLFS3Role",
            "LastModified": "2022-07-29T16:22:03.211000+00:00"
```

```
        }
    ]
}
```

자세한 내용은 Lake Formation 개발자 안내서의 Lake Formation 권한 관리를 참조하세요. AWS

- 자세한 API 내용은 명령 참조ListResources의 섹션을 참조하세요. AWS CLI

### `list-transactions`

다음 코드 예시에서는 `list-transactions`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 트랜잭션 세부 정보를 나열하려면

다음 `list-transactions` 예제에서는 트랜잭션 및 해당 상태에 대한 메타데이터를 반환합니다.

```
aws lakeformation list-transactions \
    --cli-input-json file://input.json
```

`input.json`의 콘텐츠:

```
{
    "CatalogId": "123456789111",
    "StatusFilter": "ALL",
    "MaxResults": 3
}
```

출력:

```
{
    "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+X2lhHsZLGNCDK3nNGpwRdFpicSOHgcX1/
QMoniUAKcpR3kj3ts3PVdMA=="
}
```

자세한 내용은 AWS Lake Formation 개발자 안내서의 트랜잭션 내에서 데이터 레이크에서 읽기 및 쓰기를 참조하세요.

- 자세한 API 내용은 명령 참조ListTransactions의 섹션을 참조하세요. AWS CLI

### put-data-lake-settings

다음 코드 예시에서는 put-data-lake-settings을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS Lake Formation 관리형 데이터 레이크 설정을 설정하려면

다음 put-data-lake-settings 예제에서는 데이터 레이크 관리자 및 기타 데이터 레이크 설정 목록을 설정합니다.

```
aws lakeformation put-data-lake-settings \
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
    "DataLakeSettings": {
        "DataLakeAdmins": [{
                "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-
admin"
            }
        ],
        "CreateDatabaseDefaultPermissions": [],
        "CreateTableDefaultPermissions": [],
        "TrustedResourceOwners": [],
```

```
        "AllowExternalDataFiltering": true,
        "ExternalDataFilteringAllowList": [{
            "DataLakePrincipalIdentifier ": "123456789111"
        }],
        "AuthorizedSessionTagValueList": ["Amazon EMR"]
    }
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Lake Formation 개발자 안내서의 데이터 레이크에 대한 기본 보안 설정 변경을 참조하세요.

- 자세한 API 내용은 명령 참조PutDataLakeSettings의 섹션을 참조하세요. AWS CLI

### register-resource

다음 코드 예시에서는 register-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 서비스 연결 역할을 사용하여 데이터 레이크 스토리지 등록

다음 register-resource 예제에서는 서비스 연결 역할을 사용하여 Lake Formation에서 관리하는 대로 리소스를 등록합니다.

```
aws lakeformation register-resource \
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
    "ResourceArn": "arn:aws:s3:::lf-emr-athena-result-123",
    "UseServiceLinkedRole": true
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Lake Formation 개발자 안내서의 데이터 레이크에 Amazon S3 위치 추가를 참조하세요.

예제 2: 사용자 지정 역할을 사용하여 데이터 레이크 스토리지 등록

다음 `register-resource` 예제에서는 사용자 지정 역할을 사용하여 Lake Formation에서 관리하는 대로 리소스를 등록합니다.

```
aws lakeformation register-resource \
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
    "ResourceArn": "arn:aws:s3:::lf-emr-athena-result-123",
    "UseServiceLinkedRole": false,
    "RoleArn": "arn:aws:iam::123456789111:role/LF-GlueServiceRole"
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Lake Formation 개발자 안내서의 데이터 레이크에 Amazon S3 위치 추가를 참조하세요.

- 자세한 API 내용은 명령 참조RegisterResource의 섹션을 참조하세요. AWS CLI

### remove-lf-tags-from-resource

다음 코드 예시에서는 `remove-lf-tags-from-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에서 LF 태그를 제거하려면

다음 `remove-lf-tags-from-resource` 예제에서는 테이블 리소스와의 LF 태그 연결을 제거합니다.

```
aws lakeformation remove-lf-tags-from-resource \
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
    "CatalogId": "123456789111",
    "Resource": {
        "Table": {
            "CatalogId": "123456789111",
```

```
            "DatabaseName": "tpc",
            "Name": "dl_tpc_promotion"
        }
    },
    "LFTags": [{
        "CatalogId": "123456789111",
        "TagKey": "usergroup",
        "TagValues": [
            "developer"
        ]
    }]
}
```

출력:

```
{
    "Failures": []
}
```

자세한 내용은 AWS Lake Formation 개발자 안내서의 데이터 카탈로그 리소스에 LF 태그 할당을 참조하세요.

- 자세한 API 내용은 명령 참조RemoveLfTagsFromResource의 섹션을 참조하세요. AWS CLI

### `revoke-permissions`

다음 코드 예시에서는 `revoke-permissions`을 사용하는 방법을 보여 줍니다.

AWS CLI

보안 주체의 리소스에 대한 권한을 취소하려면

다음 `revoke-permissions` 예제에서는 지정된 데이터베이스의 특정 테이블에 대한 보안 주체 액세스를 취소합니다.

```
aws lakeformation revoke-permissions \
    --cli-input-json file://input.json
```

`input.json`의 콘텐츠:

```
{
    "CatalogId": "123456789111",
```

```
    "Principal": {
        "DataLakePrincipalIdentifier": "arn:aws:iam::123456789111:user/lf-developer"
    },
    "Resource": {
        "Table": {
            "CatalogId": "123456789111",
            "DatabaseName": "tpc",
            "Name": "dl_tpc_promotion"
        }
    },
    "Permissions": [
        "ALL"
    ],
    "PermissionsWithGrantOption": []
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Lake Formation 개발자 안내서의 데이터 카탈로그 리소스에 대한 권한 부여 및 취소를 참조하세요.

- 자세한 API 내용은 명령 참조RevokePermissions의 섹션을 참조하세요. AWS CLI

### `search-databases-by-lf-tags`

다음 코드 예시에서는 `search-databases-by-lf-tags`을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터베이스 리소스를 검색하려면 LFTags

LFTag 표현식과 일치하는 데이터베이스 리소스에 대한 다음 `search-databases-by-lf-tags` 예제 검색입니다.

```
aws lakeformation search-databases-by-lf-tags \
    --cli-input-json file://input.json
```

`input.json`의 콘텐츠:

```
{
    "MaxResults": 1,
    "CatalogId": "123456789111",
    "Expression": [{
```

```
        "TagKey": "usergroup",
        "TagValues": [
            "developer"
        ]
    }]
}
```

출력:

```
{
    "DatabaseList": [{
        "Database": {
            "CatalogId": "123456789111",
            "Name": "tpc"
        },
        "LFTags": [{
            "CatalogId": "123456789111",
            "TagKey": "usergroup",
            "TagValues": [
                "developer"
            ]
        }]
    }]
}
```

자세한 내용은 AWS Lake Formation 개발자 안내서의 LF 태그가 할당된 리소스 보기를 참조하세요.

- 자세한 API 내용은 명령 참조SearchDatabasesByLfTags의 섹션을 참조하세요. AWS CLI

### `search-tables-by-lf-tags`

다음 코드 예시에서는 `search-tables-by-lf-tags`을 사용하는 방법을 보여 줍니다.

AWS CLI

를 기준으로 테이블 리소스를 검색하려면 LFTags

LFTag 표현식과 일치하는 테이블 리소스에 대한 다음 `search-tables-by-lf-tags` 예제 검색입니다.

```
aws lakeformation search-tables-by-lf-tags \
```

```
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
    "MaxResults": 2,
    "CatalogId": "123456789111",
    "Expression": [{
        "TagKey": "usergroup",
        "TagValues": [
            "developer"
        ]
    }]
}
```

출력:

```
{
    "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"
                    ]
                }]
            }
        ]
    }]
}
```

자세한 내용은 AWS Lake Formation 개발자 안내서의 LF 태그가 할당된 리소스 보기를 참조하세요.

- 자세한 API 내용은 명령 참조SearchTablesByLfTags의 섹션을 참조하세요. AWS CLI

### `start-query-planning`

다음 코드 예시에서는 `start-query-planning`을 사용하는 방법을 보여 줍니다.

AWS CLI

쿼리 문을 처리하려면

다음 `start-query-planning` 예제에서는 쿼리 문을 처리하라는 요청을 제출합니다.

```
aws lakeformation start-query-planning \
    --cli-input-json file://input.json
```

`input.json`의 콘텐츠:

```
{
    "QueryPlanningContext": {
        "CatalogId": "012345678901",
        "DatabaseName": "tpc"
    },
    "QueryString": "select * from dl_tpc_household_demographics_gov where
 hd_income_band_sk=9"
}
```

출력:

```
{
    "QueryId": "772a273f-4a62-4cda-8d98-69615ee8be9b"
}
```

자세한 내용은 AWS Lake Formation 개발자 안내서의 트랜잭션 내에서 데이터 레이크에서 읽기 및 쓰기를 참조하세요.

- 자세한 API 내용은 명령 참조StartQueryPlanning의 섹션을 참조하세요. AWS CLI

### start-transaction

다음 코드 예시에서는 start-transaction을 사용하는 방법을 보여 줍니다.

AWS CLI

새 트랜잭션을 시작하려면

다음 start-transaction 예제에서는 새 트랜잭션을 시작하고 트랜잭션 ID를 반환합니다.

```
aws lakeformation start-transaction \
    --transaction-type = 'READ_AND_WRITE'
```

출력:

```
{
    "TransactionId": "b014d972ca8347b89825e33c5774aec4"
}
```

자세한 내용은 AWS Lake Formation 개발자 안내서의 트랜잭션 내에서 데이터 레이크에서 읽기 및 쓰기를 참조하세요.

- 자세한 API 내용은 명령 참조StartTransaction의 섹션을 참조하세요. AWS CLI

### update-lf-tag

다음 코드 예시에서는 update-lf-tag을 사용하는 방법을 보여 줍니다.

AWS CLI

LF 태그 정의를 업데이트하려면

다음 update-lf-tag 예제에서는 LF 태그 정의를 업데이트합니다.

```
aws lakeformation update-lf-tag \
    --catalog-id '123456789111' \
    --tag-key 'usergroup' \
    --tag-values-to-add '["admin"]'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Lake Formation 개발자 안내서의 메타데이터 액세스 제어를 위한 LF 태그 관리를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateLfTag의 섹션을 참조하세요. AWS CLI

### update-table-objects

다음 코드 예시에서는 update-table-objects을 사용하는 방법을 보여 줍니다.

AWS CLI

관리 테이블의 객체를 수정하려면

다음 update-table-objects 예제에서는 제공된 S3 객체를 지정된 관리 테이블에 추가합니다.

```
aws lakeformation update-table-objects \
    --cli-input-json file://input.json
```

input.json의 콘텐츠:

```
{
    "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
        }
    }]
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Lake Formation 개발자 안내서의 트랜잭션 내에서 데이터 레이크에서 읽기 및 쓰기를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateTableObjects의 섹션을 참조하세요. AWS CLI

# 를 사용한 Lambda 예제 AWS CLI

다음 코드 예제에서는 Lambda와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `add-layer-version-permission`

다음 코드 예시에서는 `add-layer-version-permission`을 사용하는 방법을 보여 줍니다.

AWS CLI

계층 버전에 권한을 추가하려면

다음 `add-layer-version-permission` 예제에서는 지정된 계정이 계층 의 버전 1을 사용할 수 있는 권한을 부여합니다`my-layer`.

```
aws lambda add-layer-version-permission \
    --layer-name my-layer \
    --statement-id xaccount \
    --action lambda:GetLayerVersion  \
    --principal 123456789012 \
    --version-number 1
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS Lambda 개발자 안내서의 Lambda 계층AWS 을 참조하세요.

- 자세한 API 내용은 명령 참조AddLayerVersionPermission의 섹션을 참조하세요. AWS CLI

### add-permission

다음 코드 예시에서는 add-permission을 사용하는 방법을 보여 줍니다.

AWS CLI

기존 Lambda 함수에 권한을 추가하려면

다음 add-permission 예제에서는 Amazon SNS 서비스에 라는 함수를 호출할 수 있는 권한을 부여합니다my-function.

```
aws lambda add-permission \
    --function-name my-function \
    --action lambda:InvokeFunction \
    --statement-id sns \
    --principal sns.amazonaws.com
```

출력:

```
{
    "Statement":
    {
        "Sid":"sns",
        "Effect":"Allow",
        "Principal":{
            "Service":"sns.amazonaws.com"
        },
        "Action":"lambda:InvokeFunction",
        "Resource":"arn:aws:lambda:us-east-2:123456789012:function:my-function"
```

```
    }
}
```

자세한 내용은 AWS Lambda 개발자 안내서의 Lambda에 대한 리소스 기반 정책 사용을 참조하세요. AWS

- 자세한 API 내용은 명령 참조AddPermission의 섹션을 참조하세요. AWS CLI

### `create-alias`

다음 코드 예시에서는 `create-alias`을 사용하는 방법을 보여 줍니다.

AWS CLI

Lambda 함수에 대한 별칭을 생성하려면 다음을 수행합니다.

다음 `create-alias` 예제에서는 `my-function` Lambda 함수의 버전 1을 가리키는 `LIVE`라는 별칭을 생성합니다.

```
aws lambda create-alias \
    --function-name my-function \
    --description "alias for live version of function" \
    --function-version 1 \
    --name LIVE
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS Lambda 개발자 안내서의 Lambda 함수 별칭 구성을 참조하세요. AWS

- 자세한 API 내용은 명령 참조CreateAlias의 섹션을 참조하세요. AWS CLI

### `create-event-source-mapping`

다음 코드 예시에서는 `create-event-source-mapping`을 사용하는 방법을 보여 줍니다.

AWS CLI

이벤트 소스와 AWS Lambda 함수 간의 매핑을 생성하려면

다음 `create-event-source-mapping` 예제에서는 SQS 대기열과 `my-function` Lambda 함수 간에 매핑을 생성합니다.

```
aws lambda create-event-source-mapping \
    --function-name my-function \
    --batch-size 5 \
    --event-source-arn arn:aws:sqs:us-west-2:123456789012:mySQSqueue
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS Lambda 개발자 안내서의 Lambda 이벤트 소스 매핑을 참조하세요. AWS

- 자세한 API 내용은 명령 참조CreateEventSourceMapping의 섹션을 참조하세요. AWS CLI

### `create-function`

다음 코드 예시에서는 `create-function`을 사용하는 방법을 보여 줍니다.

AWS CLI

Lambda 함수를 생성하는 방법

다음 `create-function` 예제에서는 이름이 `my-function`인 Lambda 함수를 생성합니다.

```
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
```

my-function.zip의 콘텐츠:

```
This file is a deployment package that contains your function code and any
 dependencies.
```

출력:

```
{
    "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": ""
}
```

자세한 설명은 AWS 개발자 안내서에서 AWS Lambda 함수 구성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateFunction의 섹션을 참조하세요. AWS CLI

### delete-alias

다음 코드 예시에서는 delete-alias을 사용하는 방법을 보여 줍니다.

AWS CLI

Lambda 함수의 별칭을 삭제하려면 다음을 수행합니다.

다음 delete-alias 예제는 my-function Lambda 함수에서 LIVE라는 별칭을 삭제합니다.

```
aws lambda delete-alias \
    --function-name my-function \
    --name LIVE
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Lambda 개발자 안내서의 Lambda 함수 별칭 구성을 참조하세요. AWS

- 자세한 API 내용은 명령 참조DeleteAlias의 섹션을 참조하세요. AWS CLI

### delete-event-source-mapping

다음 코드 예시에서는 delete-event-source-mapping을 사용하는 방법을 보여 줍니다.

AWS CLI

이벤트 소스와 AWS Lambda 함수 간의 매핑을 삭제하려면

다음 delete-event-source-mapping 예제에서는 SQS 대기열과 my-function Lambda 함수 간의 매핑을 삭제합니다.

```
aws lambda delete-event-source-mapping \
    --uuid  a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS Lambda 개발자 안내서의 Lambda 이벤트 소스 매핑을 참조하세요. AWS

- 자세한 API 내용은 명령 참조DeleteEventSourceMapping의 섹션을 참조하세요. AWS CLI

### delete-function-concurrency

다음 코드 예시에서는 delete-function-concurrency을 사용하는 방법을 보여 줍니다.

AWS CLI

함수에서 예약된 동시 실행 제한을 제거하려면 다음을 수행합니다.

다음 delete-function-concurrency 예제에서는 my-function 함수에서 예약된 동시 실행 제한을 삭제합니다.

```
aws lambda delete-function-concurrency \
    --function-name  my-function
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Lambda 개발자 안내서의 Lambda 함수에 대한 동시성 예약을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteFunctionConcurrency의 섹션을 참조하세요. AWS CLI

### delete-function-event-invoke-config

다음 코드 예시에서는 delete-function-event-invoke-config을 사용하는 방법을 보여 줍니다.

AWS CLI

비동기 호출 구성을 삭제하려면

다음 delete-function-event-invoke-config 예제에서는 지정된 함수의 별GREEN칭에 대한 비동기 호출 구성을 삭제합니다.

```
aws lambda delete-function-event-invoke-config --function-name my-function:GREEN
```

- 자세한 API 내용은 명령 참조DeleteFunctionEventInvokeConfig의 섹션을 참조하세요. AWS CLI

### delete-function

다음 코드 예시에서는 delete-function을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 함수 이름을 기준으로 Lambda 함수를 삭제하는 방법

다음 delete-function 예제에서는 함수 이름을 지정하여 이름이 my-function인 Lambda 함수를 삭제합니다.

```
aws lambda delete-function \
    --function-name my-function
```

이 명령은 출력을 생성하지 않습니다.

예제 2: 함수별로 Lambda 함수를 삭제하려면 ARN

다음 delete-function 예제에서는 함수의 를 지정my-function하여 라는 Lambda 함수를 삭제합니다ARN.

```
aws lambda delete-function \
    --function-name arn:aws:lambda:us-west-2:123456789012:function:my-function
```

이 명령은 출력을 생성하지 않습니다.

예제 3: 부분 함수별로 Lambda 함수를 삭제하려면 ARN

다음 delete-function 예제에서는 함수의 부분 를 지정my-function하여 라는 Lambda 함수를 삭제합니다ARN.

```
aws lambda delete-function \
    --function-name 123456789012:function:my-function
```

이 명령은 출력을 생성하지 않습니다.

자세한 설명은 AWS 개발자 안내서에서 AWS Lambda 함수 구성을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteFunction의 섹션을 참조하세요. AWS CLI

## delete-layer-version

다음 코드 예시에서는 delete-layer-version을 사용하는 방법을 보여 줍니다.

AWS CLI

Lambda 계층의 버전을 삭제하려면

다음 delete-layer-version 예제에서는 라는 계층의 버전 2를 삭제합니다my-layer.

```
aws lambda delete-layer-version \
    --layer-name my-layer \
    --version-number 2
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Lambda 개발자 안내서의 Lambda 계층AWS 을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteLayerVersion의 섹션을 참조하세요. AWS CLI

### delete-provisioned-concurrency-config

다음 코드 예시에서는 delete-provisioned-concurrency-config을 사용하는 방법을 보여 줍니다.

AWS CLI

프로비저닝된 동시성 구성을 삭제하려면 다음을 수행합니다.

다음 delete-provisioned-concurrency-config 예제에서는 지정된 함수의 GREEN 별칭에 대해 프로비저닝된 동시성 구성을 삭제합니다.

```
aws lambda delete-provisioned-concurrency-config \
    --function-name my-function \
    --qualifier GREEN
```

- 자세한 API 내용은 명령 참조DeleteProvisionedConcurrencyConfig의 섹션을 참조하세요. AWS CLI

### get-account-settings

다음 코드 예시에서는 get-account-settings을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 리전에서 계정에 대한 세부 정보를 검색하려면

다음 get-account-settings 예제는 계정에 대한 Lambda 한도 및 사용량 정보를 표시합니다.

```
aws lambda get-account-settings
```

출력:

```
{
    "AccountLimit": {
       "CodeSizeUnzipped": 262144000,
       "UnreservedConcurrentExecutions": 1000,
       "ConcurrentExecutions": 1000,
       "CodeSizeZipped": 52428800,
       "TotalCodeSize": 80530636800
    },
    "AccountUsage": {
       "FunctionCount": 4,
       "TotalCodeSize": 9426
    }
}
```

자세한 내용은 AWS Lambda 개발자 안내서의 AWS Lambda 제한을 참조하세요.

- 자세한 API 내용은 명령 참조GetAccountSettings의 섹션을 참조하세요. AWS CLI

### get-alias

다음 코드 예시에서는 get-alias을 사용하는 방법을 보여 줍니다.

AWS CLI

함수 별칭에 대한 세부 정보를 검색하려면 다음을 수행합니다.

다음 get-alias 예제에서는 my-function Lambda 함수에서 LIVE라는 별칭에 대한 세부 정보를 표시합니다.

```
aws lambda get-alias \
    --function-name my-function \
    --name LIVE
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS Lambda 개발자 안내서의 Lambda 함수 별칭 구성을 참조하세요. AWS

- 자세한 API 내용은 명령 참조GetAlias의 섹션을 참조하세요. AWS CLI

### get-event-source-mapping

다음 코드 예시에서는 get-event-source-mapping을 사용하는 방법을 보여 줍니다.

AWS CLI

이벤트 소스 매핑에 대한 세부 정보를 검색하려면

다음 get-event-source-mapping 예제에서는 SQS 대기열과 my-function Lambda 함수 간의 매핑에 대한 세부 정보를 표시합니다.

```
aws lambda get-event-source-mapping \
    --uuid "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS Lambda 개발자 안내서의 Lambda 이벤트 소스 매핑을 참조하세요. AWS

- 자세한 API 내용은 명령 참조GetEventSourceMapping의 섹션을 참조하세요. AWS CLI

### get-function-concurrency

다음 코드 예시에서는 get-function-concurrency을 사용하는 방법을 보여 줍니다.

AWS CLI

함수에 대한 예약된 동시성 설정을 보려면 다음을 수행합니다.

다음 `get-function-concurrency` 예제에서는 지정된 함수에 대한 예약된 동시성 설정을 검색합니다.

```
aws lambda get-function-concurrency \
    --function-name my-function
```

출력:

```
{
    "ReservedConcurrentExecutions": 250
}
```

- 자세한 API 내용은 명령 참조GetFunctionConcurrency의 섹션을 참조하세요. AWS CLI

### `get-function-configuration`

다음 코드 예시에서는 `get-function-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

Lambda 함수의 버전별 설정을 검색하려면 다음을 수행합니다.

다음 `get-function-configuration` 예제에서는 `my-function` 함수의 버전 2에 대한 설정을 표시합니다.

```
aws lambda get-function-configuration \
    --function-name  my-function:2
```

출력:

```
{
    "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-uy3l9qyq",
    "Timeout": 3,
```

```
    "Runtime": "nodejs10.x",
    "TracingConfig": {
        "Mode": "PassThrough"
    },
    "CodeSha256": "5tT2qgzYUHaqwR716pZ2dpkn/0J1FrzJmlKidWoaCgk=",
    "Description": "",
    "VpcConfig": {
        "SubnetIds": [],
        "VpcId": "",
        "SecurityGroupIds": []
    },
    "CodeSize": 304,
    "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function:2",
    "Handler": "index.handler"
}
```

자세한 설명은 AWS 개발자 안내서에서 AWS Lambda 함수 구성을 참조하세요.

- 자세한 API 내용은 명령 참조GetFunctionConfiguration의 섹션을 참조하세요. AWS CLI

### get-function-event-invoke-config

다음 코드 예시에서는 get-function-event-invoke-config을 사용하는 방법을 보여 줍니다.

AWS CLI

비동기 호출 구성을 보려면

다음 get-function-event-invoke-config 예제에서는 지정된 함수의 별BLUE칭에 대한 비동기 호출 구성을 검색합니다.

```
aws lambda get-function-event-invoke-config \
    --function-name my-function:BLUE
```

출력:

```
{
    "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"
        }
    }
}
```

- 자세한 API 내용은 명령 참조GetFunctionEventInvokeConfig의 섹션을 참조하세요. AWS CLI

### get-function

다음 코드 예시에서는 get-function을 사용하는 방법을 보여 줍니다.

AWS CLI

함수 정보를 검색하는 방법

다음 get-function 예제에서는 my-function 함수에 대한 정보를 표시합니다.

```
aws lambda get-function \
    --function-name  my-function
```

출력:

```
{
    "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-
uy3l9qyq",
        "Timeout": 3,
        "LastModified": "2019-09-24T18:20:35.054+0000",
        "Runtime": "nodejs10.x",
        "Description": ""
    }
}
```

자세한 설명은 AWS 개발자 안내서에서 AWS Lambda 함수 구성을 참조하세요.

- 자세한 API 내용은 명령 참조GetFunction의 섹션을 참조하세요. AWS CLI

### get-layer-version-by-arn

다음 코드 예시에서는 get-layer-version-by-arn을 사용하는 방법을 보여 줍니다.

AWS CLI

Lambda 계층 버전에 대한 정보를 검색하려면

다음 get-layer-version-by-arn 예제에서는 지정된 Amazon 리소스 이름()이 있는 계층 버전에 대한 정보를 표시합니다ARN.

```
aws lambda get-layer-version-by-arn \
    --arn "arn:aws:lambda:us-west-2:123456789012:layer:AWSLambda-Python311-
SciPy1x:2"
```

출력:

```
{
    "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",
```

```
    "CreatedDate": "2023-10-12T10:09:38.398+0000",
    "LayerArn": "arn:aws:lambda:us-west-2:123456789012:layer:AWSLambda-Python311-
SciPy1x",
    "Content": {
        "CodeSize": 41784542,
        "CodeSha256": "GGmv8ocUw4cly0T8HL0Vx/f5V4RmSCGNjDIslY4VskM=",
        "Location": "https://awslambda-us-west-2-layers.s3.us-west-2.amazonaws.com/
snapshots/123456789012/..."
    },
    "Version": 2,
    "CompatibleRuntimes": [
        "python3.11"
    ],
    "LicenseInfo": "SciPy: https://github.com/scipy/scipy/blob/main/LICENSE.txt,
 NumPy: https://github.com/numpy/numpy/blob/main/LICENSE.txt"
}
```

자세한 내용은 AWS Lambda 개발자 안내서의 Lambda 계층AWS 을 참조하세요.

- 자세한 API 내용은 명령 참조GetLayerVersionByArn의 섹션을 참조하세요. AWS CLI

### get-layer-version-policy

다음 코드 예시에서는 get-layer-version-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

Lambda 계층 버전에 대한 권한 정책을 검색하려면

다음 get-layer-version-policy 예제에서는 라는 계층의 버전 1에 대한 정책 정보를 보여줍니다my-layer.

```
aws lambda get-layer-version-policy \
    --layer-name my-layer \
    --version-number 1
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS Lambda 개발자 안내서의 Lambda 계층AWS 을 참조하세요.

- 자세한 API 내용은 명령 참조GetLayerVersionPolicy의 섹션을 참조하세요. AWS CLI

### get-layer-version

다음 코드 예시에서는 get-layer-version을 사용하는 방법을 보여 줍니다.

AWS CLI

Lambda 계층 버전에 대한 정보를 검색하려면

다음 get-layer-version 예제에서는 라는 계층의 버전 1에 대한 정보를 표시합니다my-layer.

```
aws lambda get-layer-version \
    --layer-name my-layer \
    --version-number 1
```

출력:

```
{
    "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": "tv9jJO+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",
    "CreatedDate": "2018-11-14T23:03:52.894+0000",
    "Version": 1,
    "LicenseInfo": "MIT",
    "CompatibleRuntimes": [
        "python3.10",
        "python3.11"
    ]
}
```

자세한 내용은 AWS Lambda 개발자 안내서의 Lambda 계층AWS 을 참조하세요.

- 자세한 API 내용은 명령 참조GetLayerVersion의 섹션을 참조하세요. AWS CLI

### get-policy

다음 코드 예시에서는 get-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

함수, 버전 또는 별칭에 대한 리소스 기반 IAM 정책을 검색하려면

다음 get-policy 예제에서는 my-function Lambda 함수에 대한 정책 정보를 표시합니다.

```
aws lambda get-policy \
    --function-name my-function
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS Lambda 개발자 안내서의 Lambda에 대한 리소스 기반 정책 사용을 참조하세요. AWS

- 자세한 API 내용은 명령 참조GetPolicy의 섹션을 참조하세요. AWS CLI

### get-provisioned-concurrency-config

다음 코드 예시에서는 get-provisioned-concurrency-config을 사용하는 방법을 보여 줍니다.

AWS CLI

프로비저닝된 동시성 구성을 보려면 다음을 수행합니다.

다음 get-provisioned-concurrency-config 예제에서는 지정된 함수의 BLUE 별칭에 대해 프로비저닝된 동시성 구성의 세부 정보를 표시합니다.

```
aws lambda get-provisioned-concurrency-config \
    --function-name my-function \
    --qualifier BLUE
```

출력:

```
{
    "RequestedProvisionedConcurrentExecutions": 100,
    "AvailableProvisionedConcurrentExecutions": 100,
    "AllocatedProvisionedConcurrentExecutions": 100,
    "Status": "READY",
    "LastModified": "2019-12-31T20:28:49+0000"
}
```

- 자세한 API 내용은 명령 참조GetProvisionedConcurrencyConfig의 섹션을 참조하세요. AWS CLI

### invoke

다음 코드 예시에서는 invoke을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: Lambda 함수를 동기적으로 간접 호출하는 방법

다음 `invoke` 예제는 `my-function` 함수를 동기적으로 간접 호출합니다. 버전 2를 사용하는 AWS CLI 경우 `cli-binary-format` 옵션이 필요합니다. 자세한 내용은 AWS CLI 명령줄 인터페이스 사용 설명서의 지원되는 전역 명령줄 옵션을 참조하세요. AWS

```
aws lambda invoke \
    --function-name my-function \
    --cli-binary-format raw-in-base64-out \
    --payload '{ "name": "Bob" }' \
    response.json
```

출력:

```
{
    "ExecutedVersion": "$LATEST",
    "StatusCode": 200
}
```

자세한 내용은 AWS 개발자 안내서에서 동기식 간접 호출을 참조하세요.

예제 2: Lambda 함수를 비동기적으로 간접 호출하는 방법

다음 `invoke` 예제에서는 `my-function` 함수를 비동기적으로 간접 호출합니다. 버전 2를 사용하는 AWS CLI 경우 `cli-binary-format` 옵션이 필요합니다. 자세한 내용은 AWS CLI 명령줄 인터페이스 사용 설명서의 지원되는 전역 명령줄 옵션을 참조하세요. AWS

```
aws lambda invoke \
    --function-name my-function \
    --invocation-type Event \
    --cli-binary-format raw-in-base64-out \
    --payload '{ "name": "Bob" }' \
    response.json
```

출력:

```
{
    "StatusCode": 202
}
```

자세한 내용은 AWS 개발자 안내서에서 비동기 간접 호출을 참조하세요.

- API 자세한 내용은 AWS CLI 명령 참조의 호출을 참조하세요.

### `list-aliases`

다음 코드 예시에서는 `list-aliases`을 사용하는 방법을 보여 줍니다.

AWS CLI

Lambda 함수의 별칭 목록을 검색하려면

다음 `list-aliases` 예제에서는 `my-function` Lambda 함수의 별칭 목록을 표시합니다.

```
aws lambda list-aliases \
    --function-name my-function
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS Lambda 개발자 안내서의 Lambda 함수 별칭 구성을 참조하세요. AWS

- 자세한 API 내용은 명령 참조ListAliases의 섹션을 참조하세요. AWS CLI

### `list-event-source-mappings`

다음 코드 예시에서는 `list-event-source-mappings`을 사용하는 방법을 보여 줍니다.

AWS CLI

함수에 대한 이벤트 소스 매핑을 나열하려면

다음 `list-event-source-mappings` 예제에서는 `my-function` Lambda 함수에 대한 이벤트 소스 매핑 목록을 표시합니다.

```
aws lambda list-event-source-mappings \
    --function-name my-function
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS Lambda 개발자 안내서의 Lambda 이벤트 소스 매핑을 참조하세요. AWS

- 자세한 API 내용은 명령 참조ListEventSourceMappings의 섹션을 참조하세요. AWS CLI

### `list-function-event-invoke-configs`

다음 코드 예시에서는 `list-function-event-invoke-configs`을 사용하는 방법을 보여 줍니다.

AWS CLI

비동기 호출 구성 목록을 보려면

다음 `list-function-event-invoke-configs` 예제에서는 지정된 함수에 대한 비동기 호출 구성을 나열합니다.

```
aws lambda list-function-event-invoke-configs \
    --function-name my-function
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListFunctionEventInvokeConfigs의 섹션을 참조하세요. AWS CLI

### `list-functions`

다음 코드 예시에서는 `list-functions`을 사용하는 방법을 보여 줍니다.

AWS CLI

Lambda 함수 목록을 검색하는 방법

다음 `list-functions` 예제에서는 현재 사용자의 모든 함수 목록을 표시합니다.

```
aws lambda list-functions
```

출력:


```
{
    "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/hOZevwV/lTxCuQqK3gDZP3i8gUoqUUVRmY6E=",
            "FunctionName": "my-function",
            "VpcConfig": {
                "SubnetIds": [],
                "VpcId": "",
                "SecurityGroupIds": []
            },
            "MemorySize": 256,
            "RevisionId": "93017fc9-59cb-41dc-901b-4845ce4bf668",
            "CodeSize": 266,
            "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-
function",
            "Handler": "index.handler",
            "Role": "arn:aws:iam::123456789012:role/service-role/helloWorldPython-
role-uy3l9qyq",
```

```
            "Timeout": 3,
            "LastModified": "2023-10-01T16:47:28.490+0000",
            "Runtime": "nodejs18.x",
            "Description": ""
        },
        {
            "Layers": [
                {
                    "CodeSize": 41784542,
                    "Arn": "arn:aws:lambda:us-west-2:420165488524:layer:AWSLambda-
Python37-SciPy1x:2"
                },
                {
                    "CodeSize": 4121,
                    "Arn": "arn:aws:lambda:us-
west-2:123456789012:layer:pythonLayer:1"
                }
            ],
            "TracingConfig": {
                "Mode": "PassThrough"
            },
            "Version": "$LATEST",
            "CodeSha256": "ZQukCqxtkqFgyF2cU41Avj99TKQ/hNihPtDtRcc08mI=",
            "FunctionName": "my-python-function",
            "VpcConfig": {
                "SubnetIds": [],
                "VpcId": "",
                "SecurityGroupIds": []
            },
            "MemorySize": 128,
            "RevisionId": "80b4eabc-acf7-4ea8-919a-e874c213707d",
            "CodeSize": 299,
            "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-
python-function",
            "Handler": "lambda_function.lambda_handler",
            "Role": "arn:aws:iam::123456789012:role/service-role/my-python-function-
role-z5g7dr6n",
            "Timeout": 3,
            "LastModified": "2023-10-01T19:40:41.643+0000",
            "Runtime": "python3.11",
            "Description": ""
        }
    ]
```

```
}
```

자세한 설명은 AWS 개발자 안내서에서 AWS Lambda 함수 구성을 참조하세요.

- 자세한 API 내용은 명령 참조ListFunctions의 섹션을 참조하세요. AWS CLI

### `list-layer-versions`

다음 코드 예시에서는 `list-layer-versions`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS Lambda 계층의 버전을 나열하려면

다음 `list-layers-versions` 예제에서는 라는 계층의 버전에 대한 정보를 표시합니다`my-layer`.

```
aws lambda list-layer-versions \
    --layer-name my-layer
```

출력:

```
{
    "Layers": [
        {
            "LayerVersionArn": "arn:aws:lambda:us-east-2:123456789012:layer:my-
layer:2",
            "Version": 2,
            "Description": "My layer",
            "CreatedDate": "2023-11-15T00:37:46.592+0000",
            "CompatibleRuntimes": [
                "python3.10",
                "python3.11"
            ]
        }
    ]
}
```

자세한 내용은 AWS Lambda 개발자 안내서의 Lambda 계층AWS 을 참조하세요.

- 자세한 API 내용은 명령 참조ListLayerVersions의 섹션을 참조하세요. AWS CLI

### list-layers

다음 코드 예시에서는 list-layers을 사용하는 방법을 보여 줍니다.

AWS CLI

함수의 런타임과 호환되는 계층을 나열하려면

다음 list-layers 예제에서는 Python 3.11 런타임과 호환되는 계층에 대한 정보를 보여줍니다.

```
aws lambda list-layers \
    --compatible-runtime python3.11
```

출력:

```
{
    "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",
                "CreatedDate": "2023-11-15T00:37:46.592+0000",
                "CompatibleRuntimes": [
                    "python3.10",
                    "python3.11"
                ]
            }
        }
    ]
}
```

자세한 내용은 AWS Lambda 개발자 안내서의 Lambda 계층AWS 을 참조하세요.

- 자세한 API 내용은 명령 참조ListLayers의 섹션을 참조하세요. AWS CLI

### list-provisioned-concurrency-configs

다음 코드 예시에서는 list-provisioned-concurrency-configs을 사용하는 방법을 보여 줍니다.

AWS CLI

프로비저닝된 동시성 구성의 목록을 가져오려면 다음을 수행합니다.

다음 `list-provisioned-concurrency-configs` 예제에는 지정된 함수에 대한 프로비저닝된 동시성 구성이 나열되어 있습니다.

```
aws lambda list-provisioned-concurrency-configs \
    --function-name my-function
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListProvisionedConcurrencyConfigs의 섹션을 참조하세요. AWS CLI

### `list-tags`

다음 코드 예시에서는 `list-tags`을 사용하는 방법을 보여 줍니다.

AWS CLI

Lambda 함수에 대한 태그 목록을 검색하려면 다음을 수행합니다.

다음 `list-tags` 예제에서는 `my-function` Lambda 함수에 연결된 태그를 표시합니다.

```
aws lambda list-tags \
    --resource arn:aws:lambda:us-west-2:123456789012:function:my-function
```

출력:

```
{
    "Tags": {
        "Category": "Web Tools",
        "Department": "Sales"
    }
}
```

자세한 내용은 AWS Lambda 개발자 안내서의 Lambda 함수 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조ListTags의 섹션을 참조하세요. AWS CLI

### `list-versions-by-function`

다음 코드 예시에서는 `list-versions-by-function`을 사용하는 방법을 보여 줍니다.

AWS CLI

함수 버전의 목록을 검색하려면 다음을 수행합니다.

다음 `list-versions-by-function` 예제에서는 `my-function` Lambda 함수의 버전 목록을 표시합니다.

```
aws lambda list-versions-by-function \
    --function-name my-function
```

출력:

```
{
    "Versions": [
        {
            "TracingConfig": {
```

```
                "Mode": "PassThrough"
            },
            "Version": "$LATEST",
            "CodeSha256": "sU0cJ2/hOZevwV/lTxCuQqK3gDZP3i8gUoqUUVRmY6E=",
            "FunctionName": "my-function",
            "VpcConfig": {
                "SubnetIds": [],
                "VpcId": "",
                "SecurityGroupIds": []
            },
            "MemorySize": 256,
            "RevisionId": "93017fc9-59cb-41dc-901b-4845ce4bf668",
            "CodeSize": 266,
            "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-
function:$LATEST",
            "Handler": "index.handler",
            "Role": "arn:aws:iam::123456789012:role/service-role/helloWorldPython-
role-uy3l9qyq",
            "Timeout": 3,
            "LastModified": "2019-10-01T16:47:28.490+0000",
            "Runtime": "nodejs10.x",
            "Description": ""
        },
        {
            "TracingConfig": {
                "Mode": "PassThrough"
            },
            "Version": "1",
            "CodeSha256": "5tT2qgzYUHoqwR616pZ2dpkn/0J1FrzJmlKidWaaCgk=",
            "FunctionName": "my-function",
            "VpcConfig": {
                "SubnetIds": [],
                "VpcId": "",
                "SecurityGroupIds": []
            },
            "MemorySize": 256,
            "RevisionId": "949c8914-012e-4795-998c-e467121951b1",
            "CodeSize": 304,
            "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-
function:1",
            "Handler": "index.handler",
            "Role": "arn:aws:iam::123456789012:role/service-role/helloWorldPython-
role-uy3l9qyq",
            "Timeout": 3,
```

```
            "LastModified": "2019-09-26T20:28:40.438+0000",
            "Runtime": "nodejs10.x",
            "Description": "new version"
        },
        {
            "TracingConfig": {
                "Mode": "PassThrough"
            },
            "Version": "2",
            "CodeSha256": "sU0cJ2/hOZevwV/lTxCuQqK3gDZP3i8gUoqUUVRmY6E=",
            "FunctionName": "my-function",
            "VpcConfig": {
                "SubnetIds": [],
                "VpcId": "",
                "SecurityGroupIds": []
            },
            "MemorySize": 256,
            "RevisionId": "cd669f21-0f3d-4e1c-9566-948837f2e2ea",
            "CodeSize": 266,
            "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-
function:2",
            "Handler": "index.handler",
            "Role": "arn:aws:iam::123456789012:role/service-role/helloWorldPython-
role-uy3l9qyq",
            "Timeout": 3,
            "LastModified": "2019-10-01T16:47:28.490+0000",
            "Runtime": "nodejs10.x",
            "Description": "newer version"
        }
    ]
}
```

자세한 내용은 AWS Lambda 개발자 안내서의 Lambda 함수 별칭 구성을 참조하세요. AWS

- 자세한 API 내용은 명령 참조ListVersionsByFunction의 섹션을 참조하세요. AWS CLI

### `publish-layer-version`

다음 코드 예시에서는 `publish-layer-version`을 사용하는 방법을 보여 줍니다.

AWS CLI

Lambda 계층 버전을 생성하려면

다음 publish-layer-version 예제에서는 새 Python 라이브러리 계층 버전을 생성합니다. 명령은 지정된 S3 버킷layer.zip에 이름이 지정된 파일인 계층 콘텐츠를 검색합니다.

```
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
```

출력:

```
{
    "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": "tv9jJO+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",
    "CreatedDate": "2023-11-14T23:03:52.894+0000",
    "Version": 1,
    "LicenseInfo": "MIT",
    "CompatibleRuntimes": [
        "python3.10",
        "python3.11"
    ]
}
```

자세한 내용은 AWS Lambda 개발자 안내서의 Lambda 계층AWS 을 참조하세요.

- 자세한 API 내용은 명령 참조PublishLayerVersion의 섹션을 참조하세요. AWS CLI

### publish-version

다음 코드 예시에서는 publish-version을 사용하는 방법을 보여 줍니다.

AWS CLI

함수의 새 버전을 게시하려면 다음을 수행합니다.

다음 publish-version 예제에서는 my-function Lambda 함수의 새 버전을 게시합니다.

```
aws lambda publish-version \
    --function-name my-function
```

출력:

```
{
    "TracingConfig": {
        "Mode": "PassThrough"
    },
    "CodeSha256": "dBG9m8SGdmlEjw/JYXlhhvCrAv5TxvXsbL/RMr0fT/I=",
    "FunctionName": "my-function",
    "CodeSize": 294,
    "RevisionId": "f31d3d39-cc63-4520-97d4-43cd44c94c20",
    "MemorySize": 128,
    "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function:3",
    "Version": "2",
    "Role": "arn:aws:iam::123456789012:role/service-role/MyTestFunction-role-
zgur6bf4",
    "Timeout": 3,
    "LastModified": "2019-09-23T18:32:33.857+0000",
    "Handler": "my-function.handler",
    "Runtime": "nodejs10.x",
    "Description": ""
}
```

자세한 내용은 AWS Lambda 개발자 안내서의 Lambda 함수 별칭 구성을 참조하세요. AWS

- 자세한 API 내용은 명령 참조PublishVersion의 섹션을 참조하세요. AWS CLI

### put-function-concurrency

다음 코드 예시에서는 put-function-concurrency을 사용하는 방법을 보여 줍니다.

AWS CLI

함수에 대해 예약된 동시성 한도를 구성하려면 다음을 수행합니다.

다음 put-function-concurrency 예제에서는 my-function 함수에 대해 100개의 예약된 동시 실행을 구성합니다.

```
aws lambda put-function-concurrency \
```

```
    --function-name  my-function  \
    --reserved-concurrent-executions 100
```

출력:

```
{
    "ReservedConcurrentExecutions": 100
}
```

자세한 내용은 AWS Lambda 개발자 안내서의 Lambda 함수에 대한 동시성 예약을 참조하세요.

- 자세한 API 내용은 명령 참조PutFunctionConcurrency의 섹션을 참조하세요. AWS CLI

### put-function-event-invoke-config

다음 코드 예시에서는 put-function-event-invoke-config을 사용하는 방법을 보여 줍니다.

AWS CLI

비동기 호출에 대한 오류 처리를 구성하려면

다음 put-function-event-invoke-config 예제에서는 최대 이벤트 기간을 1시간으로 설정하고 지정된 함수에 대한 재시도를 비활성화합니다.

```
aws lambda put-function-event-invoke-config \
    --function-name my-function \
    --maximum-event-age-in-seconds 3600 \
    --maximum-retry-attempts 0
```

출력:

```
{
    "LastModified": 1573686021.479,
    "FunctionArn": "arn:aws:lambda:us-east-2:123456789012:function:my-function:
$LATEST",
    "MaximumRetryAttempts": 0,
    "MaximumEventAgeInSeconds": 3600,
    "DestinationConfig": {
        "OnSuccess": {},
        "OnFailure": {}
    }
}
```

- 자세한 API 내용은 명령 참조PutFunctionEventInvokeConfig의 섹션을 참조하세요. AWS CLI

### put-provisioned-concurrency-config

다음 코드 예시에서는 put-provisioned-concurrency-config을 사용하는 방법을 보여 줍니다.

AWS CLI

프로비저닝된 동시성을 할당하려면 다음을 수행합니다.

다음 put-provisioned-concurrency-config 예제에서는 지정된 함수의 BLUE 별칭에 대해 프로비저닝된 동시성 100개를 할당합니다.

```
aws lambda put-provisioned-concurrency-config \
    --function-name my-function \
    --qualifier BLUE \
    --provisioned-concurrent-executions 100
```

출력:

```
{
    "Requested ProvisionedConcurrentExecutions": 100,
    "Allocated ProvisionedConcurrentExecutions": 0,
    "Status": "IN_PROGRESS",
    "LastModified": "2019-11-21T19:32:12+0000"
}
```

- 자세한 API 내용은 명령 참조PutProvisionedConcurrencyConfig의 섹션을 참조하세요. AWS CLI

### remove-layer-version-permission

다음 코드 예시에서는 remove-layer-version-permission을 사용하는 방법을 보여 줍니다.

AWS CLI

계층 버전 권한을 삭제하려면

다음 remove-layer-version-permission 예제에서는 계정이 계층 버전을 구성할 수 있는 권한을 삭제합니다.

```
aws lambda remove-layer-version-permission \
```

```
    --layer-name my-layer \
    --statement-id xaccount \
    --version-number 1
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Lambda 개발자 안내서의 Lambda 계층AWS 을 참조하세요.

- 자세한 API 내용은 명령 참조RemoveLayerVersionPermission의 섹션을 참조하세요. AWS CLI

### `remove-permission`

다음 코드 예시에서는 `remove-permission`을 사용하는 방법을 보여 줍니다.

AWS CLI

기존 Lambda 함수에서 권한을 제거하려면 다음을 수행합니다.

다음 `remove-permission` 예제에서는 `my-function`이라는 함수를 간접적으로 호출할 수 있는 권한을 제거합니다.

```
aws lambda remove-permission \
    --function-name my-function \
    --statement-id sns
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Lambda 개발자 안내서의 Lambda에 대한 리소스 기반 정책 사용을 참조하세요. AWS

- 자세한 API 내용은 명령 참조RemovePermission의 섹션을 참조하세요. AWS CLI

### `tag-resource`

다음 코드 예시에서는 `tag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

기존 Lambda 함수에 태그를 추가하려면 다음을 수행합니다.

다음 `tag-resource` 예제는 지정된 Lambda 함수에 키 이름 `DEPARTMENT`와 값이 `Department A`인 태그를 추가합니다.

```
aws lambda tag-resource \
    --resource arn:aws:lambda:us-west-2:123456789012:function:my-function \
    --tags "DEPARTMENT=Department A"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Lambda 개발자 안내서의 Lambda 함수 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### untag-resource

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

기존 Lambda 함수에서 태그를 제거하려면 다음을 수행합니다.

다음 untag-resource 예제에서는 my-function Lambda 함수에서 키 이름 DEPARTMENT 태그가 있는 태그를 제거합니다.

```
aws lambda untag-resource \
    --resource arn:aws:lambda:us-west-2:123456789012:function:my-function \
    --tag-keys DEPARTMENT
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Lambda 개발자 안내서의 Lambda 함수 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-alias

다음 코드 예시에서는 update-alias을 사용하는 방법을 보여 줍니다.

AWS CLI

함수 별칭을 업데이트하려면 다음을 수행합니다.

다음 update-alias 예제에서는 my-function Lambda 함수의 버전 3을 가리키도록 LIVE라는 별칭을 업데이트합니다.

```
aws lambda update-alias \
    --function-name my-function \
    --function-version 3 \
    --name LIVE
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS Lambda 개발자 안내서의 Lambda 함수 별칭 구성을 참조하세요. AWS

- 자세한 API 내용은 명령 참조UpdateAlias의 섹션을 참조하세요. AWS CLI

### update-event-source-mapping

다음 코드 예시에서는 update-event-source-mapping을 사용하는 방법을 보여 줍니다.

AWS CLI

이벤트 소스와 AWS Lambda 함수 간의 매핑을 업데이트하려면

다음 update-event-source-mapping 예제에서는 지정된 매핑에서 배치 크기를 8로 업데이트합니다.

```
aws lambda update-event-source-mapping \
    --uuid  "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE" \
    --batch-size 8
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS Lambda 개발자 안내서의 Lambda 이벤트 소스 매핑을 참조하세요. AWS

- 자세한 API 내용은 명령 참조UpdateEventSourceMapping의 섹션을 참조하세요. AWS CLI

### update-function-code

다음 코드 예시에서는 update-function-code을 사용하는 방법을 보여 줍니다.

AWS CLI

Lambda 함수 코드를 업데이트하는 방법

다음 update-function-code 예제에서는 my-function 함수의 게시되지 않은($LATEST) 버전의 코드를 지정된 zip 파일의 내용으로 바꿉니다.

```
aws lambda update-function-code \
    --function-name  my-function \
    --zip-file fileb://my-function.zip
```

출력:

```
{
    "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-uy3l9qyq",
    "Timeout": 3,
    "Runtime": "nodejs10.x",
    "TracingConfig": {
        "Mode": "PassThrough"
    },
    "CodeSha256": "5tT2qgzYUHaqwR716pZ2dpkn/0J1FrzJmlKidWoaCgk=",
    "Description": "",
    "VpcConfig": {
        "SubnetIds": [],
        "VpcId": "",
```

```
        "SecurityGroupIds": []
    },
    "CodeSize": 304,
    "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function",
    "Handler": "index.handler"
}
```

자세한 설명은 AWS 개발자 안내서에서 AWS Lambda 함수 구성을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateFunctionCode의 섹션을 참조하세요. AWS CLI

### update-function-configuration

다음 코드 예시에서는 update-function-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

함수 구성을 수정하는 방법

다음 update-function-configuration 예제에서는 my-function 함수의 게시되지 않은 ($LATEST) 버전에 대해 메모리 크기를 256MB로 수정합니다.

```
aws lambda update-function-configuration \
    --function-name  my-function \
    --memory-size 256
```

출력:

```
{
    "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-uy3l9qyq",
    "Timeout": 3,
    "Runtime": "nodejs10.x",
    "TracingConfig": {
        "Mode": "PassThrough"
    },
    "CodeSha256": "5tT2qgzYUHaqwR716pZ2dpkn/0J1FrzJmlKidWoaCgk=",
    "Description": "",
    "VpcConfig": {
```

```
        "SubnetIds": [],
        "VpcId": "",
        "SecurityGroupIds": []
    },
    "CodeSize": 304,
    "FunctionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-function",
    "Handler": "index.handler"
}
```

자세한 설명은 AWS 개발자 안내서에서 AWS Lambda 함수 구성을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateFunctionConfiguration의 섹션을 참조하세요. AWS CLI

### update-function-event-invoke-config

다음 코드 예시에서는 update-function-event-invoke-config을 사용하는 방법을 보여 줍니다.

AWS CLI

비동기 호출 구성을 업데이트하려면

다음 update-function-event-invoke-config 예제에서는 지정된 함수에 대한 기존 비동기 호출 구성에 실패 시 대상을 추가합니다.

```
aws lambda update-function-event-invoke-config \
    --function-name my-function \
    --destination-config '{"OnFailure":{"Destination": "arn:aws:sqs:us-
east-2:123456789012:destination"}}'
```

출력:

```
{
    "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"
```

```
        }
    }
}
```

- 자세한 API 내용은 명령 참조UpdateFunctionEventInvokeConfig의 섹션을 참조하세요. AWS CLI

## 를 사용한 License Manager 예제 AWS CLI

다음 코드 예제에서는 License Manager AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `create-license-configuration`

다음 코드 예시에서는 `create-license-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 라이선스 구성을 생성하려면

다음 `create-license-configuration` 예제에서는 코어 10개로 제한되는 라이선스 구성을 생성합니다.

```
aws license-manager create-license-configuration --name my-license-configuration \
    --license-counting-type Core \
    --license-count 10 \
    --license-count-hard-limit
```

출력:

```
{
  "LicenseConfigurationArn": "arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-6eb6586f508a786a2ba41EXAMPLE1111"
}
```

예제 2: 라이선스 구성을 생성하려면

다음 create-license-configuration 예제에서는 소프트 제한이 100인 라이선스 구성을 생성합니다vCPUs. 규칙을 사용하여 vCPU 최적화를 활성화합니다.

```
aws license-manager create-license-configuration --name my-license-configuration
    --license-counting-type vCPU \
    --license-count 100 \
    --license-rules "#honorVcpuOptimization=true"
```

출력:

```
{
  "LicenseConfigurationArn": "arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-6eb6586f508a786a2ba41EXAMPLE2222"
}
```

- 자세한 API 내용은 명령 참조CreateLicenseConfiguration의 섹션을 참조하세요. AWS CLI

### delete-license-configuration

다음 코드 예시에서는 delete-license-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

라이선스 구성을 삭제하려면

다음 delete-license-configuration 예제에서는 지정된 라이선스 구성을 삭제합니다.

```
aws license-manager delete-license-configuration \
    --license-configuration-arn arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-6eb6586f508a786a2ba4f56c1EXAMPLE
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeleteLicenseConfiguration의 섹션을 참조하세요. AWS CLI

### get-license-configuration

다음 코드 예시에서는 get-license-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

라이선스 구성 정보를 가져오려면

다음 get-license-configuration 예제에서는 지정된 라이선스 구성에 대한 세부 정보를 표시합니다.

```
aws license-manager get-license-configuration \
    --license-configuration-arn arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-38b658717b87478aaa7c00883EXAMPLE
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetLicenseConfiguration의 섹션을 참조하세요. AWS CLI

### `get-service-settings`

다음 코드 예시에서는 `get-service-settings`을 사용하는 방법을 보여 줍니다.

AWS CLI

License Manager 설정을 가져오려면

다음 `get-service-settings` 예제에서는 현재 리전의 License Manager에 대한 서비스 설정을 보여줍니다.

```
aws license-manager get-service-settings
```

다음은 교차 계정 리소스 검색이 비활성화된 경우의 예제 출력입니다.

```
{
    "OrganizationConfiguration": {
        "EnableIntegration": false
    },
    "EnableCrossAccountsDiscovery": false
}
```

다음은 교차 계정 리소스 검색이 활성화된 경우의 예제 출력입니다.

```
{
    "S3BucketArn": "arn:aws:s3:::aws-license-manager-service-c22d6279-35c4-47c4-bb",
    "OrganizationConfiguration": {
        "EnableIntegration": true
    },
    "EnableCrossAccountsDiscovery": true
}
```

- 자세한 API 내용은 명령 참조GetServiceSettings의 섹션을 참조하세요. AWS CLI

### list-associations-for-license-configuration

다음 코드 예시에서는 list-associations-for-license-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

라이선스 구성에 대한 연결을 가져오려면

다음 list-associations-for-license-configuration 예제에서는 지정된 라이선스 구성의 연결에 대한 자세한 정보를 표시합니다.

```
aws license-manager list-associations-for-license-configuration \
    --license-configuration-arn arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-38b658717b87478aaa7c00883EXAMPLE
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListAssociationsForLicenseConfiguration의 섹션을 참조하세요. AWS CLI

### `list-license-configurations`

다음 코드 예시에서는 `list-license-configurations`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 모든 라이선스 구성을 나열하려면

다음 `list-license-configurations` 예제에서는 모든 라이선스 구성을 나열합니다.

```
aws license-manager list-license-configurations
```

출력:

```
{
    "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
                }
            ]
        },
        {
            ...
        }
    ]
}
```

예제 2: 특정 라이선스 구성을 나열하려면

다음 `list-license-configurations` 예제에서는 지정된 라이선스 구성만 나열합니다.

```
aws license-manager list-license-configurations \
    --license-configuration-arns arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-38b658717b87478aaa7c00883EXAMPLE
```

- 자세한 API 내용은 명령 참조ListLicenseConfigurations의 섹션을 참조하세요. AWS CLI

### list-license-specifications-for-resource

다음 코드 예시에서는 list-license-specifications-for-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 대한 라이선스 구성을 나열하려면

다음 list-license-specifications-for-resource 예제에서는 지정된 Amazon Machine Image()와 연결된 라이선스 구성을 나열합니다AMI.

```
aws license-manager list-license-specifications-for-resource \
    --resource-arn arn:aws:ec2:us-west-2::image/ami-1234567890abcdef0
```

출력:

```
{
    "LicenseConfigurationArn": "arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-38b658717b87478aaa7c00883EXAMPLE"
}
```

- 자세한 API 내용은 명령 참조ListLicenseSpecificationsForResource의 섹션을 참조하세요. AWS CLI

### list-resource-inventory

다음 코드 예시에서는 list-resource-inventory을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 인벤토리의 리소스를 나열하려면

다음 list-resource-inventory 예제에서는 Systems Manager 인벤토리를 사용하여 관리되는 리소스를 나열합니다.

```
aws license-manager list-resource-inventory
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListResourceInventory의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

라이선스 구성의 태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 지정된 라이선스 구성의 태그를 나열합니다.

```
aws license-manager list-tags-for-resource \
```

```
    --resource-arn arn:aws:license-manager:us-west-2:123456789012:license-
configuration:lic-6eb6586f508a786a2ba4f56c1EXAMPLE
```

출력:

```
{
    "Tags": [
        {
            "Key": "project",
            "Value": "lima"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### list-usage-for-license-configuration

다음 코드 예시에서는 list-usage-for-license-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

라이선스 구성에 사용 중인 라이선스를 나열하려면

다음 list-usage-for-license-configuration 예제에서는 지정된 라이선스 구성에 대한 라이선스를 사용하는 리소스에 대한 정보를 나열합니다. 예를 들어 라이선스 유형이 v 인 경우 CPU 모든 인스턴스는 v당 하나의 라이선스를 사용합니다CPU.

```
aws license-manager list-usage-for-license-configuration \
    --license-configuration-arn arn:aws:license-manager:us-
west-2:123456789012:license-configuration:lic-38b658717b87478aaa7c00883EXAMPLE
```

출력:

```
{
    "LicenseConfigurationUsageList": [
        {
            "ResourceArn": "arn:aws:ec2:us-west-2:123456789012:instance/
i-04a636d18e83cfacb",
            "ResourceType": "EC2_INSTANCE",
```

```
            "ResourceStatus": "running",
            "ResourceOwnerId": "123456789012",
            "AssociationTime": 1570892850.519,
            "ConsumedLicenses": 2
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListUsageForLicenseConfiguration의 섹션을 참조하세요. AWS CLI

### `tag-resource`

다음 코드 예시에서는 `tag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

라이선스 구성에 태그를 추가하려면

다음 `tag-resource` 예제에서는 지정된 태그(키 이름 및 값)를 지정된 라이선스 구성에 추가합니다.

```
aws license-manager tag-resource \
    --tags Key=project,Value=lima \
    --resource-arn arn:aws:license-manager:us-west-2:123456789012:license-
configuration:lic-6eb6586f508a786a2ba4f56c1EXAMPLE
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### `untag-resource`

다음 코드 예시에서는 `untag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

라이선스 구성에서 태그를 제거하려면

다음 `untag-resource` 예제에서는 지정된 라이선스 구성에서 지정된 태그(키 이름 및 리소스)를 제거합니다.

```
aws license-manager untag-resource \
```

```
    --tag-keys project \
    --resource-arn arn:aws:license-manager:us-west-2:123456789012:license-
configuration:lic-6eb6586f508a786a2ba4f56c1EXAMPLE
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-license-configuration

다음 코드 예시에서는 update-license-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

라이선스 구성을 업데이트하려면

다음 update-license-configuration 예제에서는 지정된 라이선스 구성을 업데이트하여 하드 제한을 제거합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

다음 update-license-configuration 예제에서는 지정된 라이선스 구성을 업데이트하여 상태를 로 변경합니다DISABLED.

```
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
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조UpdateLicenseConfiguration의 섹션을 참조하세요. AWS CLI

### update-license-specifications-for-resource

다음 코드 예시에서는 update-license-specifications-for-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 대한 라이선스 구성을 업데이트하려면

다음 update-license-specifications-for-resource 예제에서는 한 라이선스 구성을 제거하고 다른 라이선스를 추가하여 지정된 Amazon Machine Image(AMI)와 연결된 라이선스 구성을 대체합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조UpdateLicenseSpecificationsForResource의 섹션을 참조하세요. AWS CLI

### update-service-settings

다음 코드 예시에서는 update-service-settings을 사용하는 방법을 보여 줍니다.

AWS CLI

License Manager 설정을 업데이트하려면

다음 update-service-settings 예제에서는 현재 AWS 리전에서 License Manager에 대한 교차 계정 리소스 검색을 활성화합니다. Amazon S3 버킷은 Systems Manager 인벤토리에 필요한 리소스 데이터 동기화입니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조UpdateServiceSettings의 섹션을 참조하세요. AWS CLI

# 를 사용한 Lightsail 예제 AWS CLI

다음 코드 예제에서는 Lightsail과 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `allocate-static-ip`

다음 코드 예시에서는 `allocate-static-ip`을 사용하는 방법을 보여 줍니다.

AWS CLI

정적 IP를 생성하려면

다음 `allocate-static-ip` 예제에서는 인스턴스에 연결할 수 있는 지정된 정적 IP를 생성합니다.

```
aws lightsail allocate-static-ip \
    --static-ip-name StaticIp-1
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조AllocateStaticIp의 섹션을 참조하세요. AWS CLI

### attach-disk

다음 코드 예시에서는 attach-disk을 사용하는 방법을 보여 줍니다.

AWS CLI

블록 스토리지 디스크를 인스턴스에 연결하려면

다음 attach-disk 예제에서는 의 디스크 경로를 WordPress_Multisite-1 사용하여 인스턴스에 디스크Disk-1를 연결합니다. /dev/xvdf

```
aws lightsail attach-disk \
    --disk-name Disk-1 \
    --disk-path /dev/xvdf \
    --instance-name WordPress_Multisite-1
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조AttachDisk의 섹션을 참조하세요. AWS CLI

### attach-instances-to-load-balancer

다음 코드 예시에서는 attach-instances-to-load-balancer을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스를 로드 밸런서에 연결하려면

다음 attach-instances-to-load-balancer 예제에서는 인스턴스 MEAN-1, MEAN-2및 MEAN-3를 로드 밸런서 에 연결합니다LoadBalancer-1.

```
aws lightsail attach-instances-to-load-balancer \
    --instance-names {"MEAN-1","MEAN-2","MEAN-3"} \
    --load-balancer-name LoadBalancer-1
```

출력:


```
{
    "operations": [
        {
            "id": "8055d19d-abb2-40b9-b527-1EXAMPLE3c7b",
            "resourceName": "LoadBalancer-1",
            "resourceType": "LoadBalancer",
            "createdAt": 1571071699.892,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "MEAN-2",
            "operationType": "AttachInstancesToLoadBalancer",
            "status": "Started",
            "statusChangedAt": 1571071699.892
        },
        {
            "id": "c35048eb-8538-456a-a118-0EXAMPLEfb73",
            "resourceName": "MEAN-2",
            "resourceType": "Instance",
            "createdAt": 1571071699.887,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "LoadBalancer-1",
            "operationType": "AttachInstancesToLoadBalancer",
            "status": "Started",
            "statusChangedAt": 1571071699.887
        },
        {
            "id": "910d09e0-adc5-4372-bc2e-0EXAMPLEd891",
            "resourceName": "LoadBalancer-1",
            "resourceType": "LoadBalancer",
            "createdAt": 1571071699.882,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
```

```
            "operationDetails": "MEAN-3",
            "operationType": "AttachInstancesToLoadBalancer",
            "status": "Started",
            "statusChangedAt": 1571071699.882
        },
        {
            "id": "178b18ac-43e8-478c-9bed-1EXAMPLE4755",
            "resourceName": "MEAN-3",
            "resourceType": "Instance",
            "createdAt": 1571071699.901,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "LoadBalancer-1",
            "operationType": "AttachInstancesToLoadBalancer",
            "status": "Started",
            "statusChangedAt": 1571071699.901
        },
        {
            "id": "fb62536d-2a98-4190-a6fc-4EXAMPLE7470",
            "resourceName": "LoadBalancer-1",
            "resourceType": "LoadBalancer",
            "createdAt": 1571071699.885,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationDetails": "MEAN-1",
            "operationType": "AttachInstancesToLoadBalancer",
            "status": "Started",
            "statusChangedAt": 1571071699.885
        },
        {
            "id": "787dac0d-f98d-46c3-8571-3EXAMPLE5a85",
            "resourceName": "MEAN-1",
            "resourceType": "Instance",
            "createdAt": 1571071699.901,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
```

```
            "isTerminal": false,
            "operationDetails": "LoadBalancer-1",
            "operationType": "AttachInstancesToLoadBalancer",
            "status": "Started",
            "statusChangedAt": 1571071699.901
        }
    ]
}
```

- 자세한 API 내용은 명령 참조AttachInstancesToLoadBalancer의 섹션을 참조하세요. AWS CLI

### attach-load-balancer-tls-certificate

다음 코드 예시에서는 attach-load-balancer-tls-certificate을 사용하는 방법을 보여 줍니다.

AWS CLI

로드 밸런서에 TLS 인증서를 연결하려면

다음 attach-load-balancer-tls-certificate 예제에서는 로드 밸런서 TLS 인증서를 로드 밸런서 Certificate2에 연결합니다LoadBalancer-1.

```
aws lightsail attach-load-balancer-tls-certificate \
    --certificate-name Certificate2 \
    --load-balancer-name LoadBalancer-1
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조AttachLoadBalancerTlsCertificate의 섹션을 참조하세요. AWS CLI

### attach-static-ip

다음 코드 예시에서는 attach-static-ip을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스에 정적 IP를 연결하려면

다음 attach-static-ip 예제에서는 정적 IP를 인스턴스 StaticIp-1에 연결합니다MEAN-1.

```
aws lightsail attach-static-ip \
    --static-ip-name StaticIp-1 \
    --instance-name MEAN-1
```

출력:

```
{
```

```
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조AttachStaticIp의 섹션을 참조하세요. AWS CLI

### close-instance-public-ports

다음 코드 예시에서는 close-instance-public-ports을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스의 방화벽 포트를 닫으려면

다음 close-instance-public-ports 예제에서는 인스턴스 22의 TCP포트를 닫습니다MEAN-2.

```
aws lightsail close-instance-public-ports \
    --instance-name MEAN-2 \
    --port-info fromPort=22,protocol=TCP,toPort=22
```

출력:

```
{
    "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
    }
}
```

- 자세한 API 내용은 명령 참조CloseInstancePublicPorts의 섹션을 참조하세요. AWS CLI

### copy-snapshot

다음 코드 예시에서는 copy-snapshot을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 동일한 AWS 리전 내에서 스냅샷을 복사하려면

다음 copy-snapshot 예제는 동일한 AWS 리전 MEAN-1-Copy 내에서 인스턴스 스냅샷을 인스턴스 스냅샷MEAN-1-1571075291으로 복사합니다us-west-2.

```
aws lightsail copy-snapshot \
```

```
    --source-snapshot-name MEAN-1-1571075291 \
    --target-snapshot-name MEAN-1-Copy \
    --source-region us-west-2
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 Lightsail 개발 가이드의 Amazon Lightsail에서 한 리전에서 다른 AWS 리전으로 스냅샷 복사를 참조하세요.

예제 2: 한 리전에서 다른 AWS 리전으로 스냅샷 복사

다음 `copy-snapshot` 예제에서는 인스턴스 스냅샷을 AWS 리전`MEAN-1-1571075291-Copy`에서 `MEAN-1-1571075291`로 인스턴스 스냅샷으로 복사`us-west-2`합니다`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
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 Lightsail 개발 가이드의 Amazon Lightsail에서 한 리전에서 다른 AWS 리전으로 스냅샷 복사를 참조하세요.

예제 3: 동일한 AWS 리전 내에서 자동 스냅샷을 복사하려면

다음 copy-snapshot 예제는 AWS 리전 WordPress-1-10142019에서 인스턴스2019-10-14의 자동 스냅샷을 수동 스냅샷WordPress-1으로 복사합니다us-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
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 Lightsail Dev Guide의 Amazon Lightsail에서 인스턴스 또는 디스크의 자동 스냅샷 유지를 참조하세요.

예제 4: 한 리전에서 다른 AWS 리전으로 자동 스냅샷 복사

다음 copy-snapshot 예제에서는 인스턴스2019-10-14의 자동 스냅샷을 AWS 리전WordPress-1-10142019에서 로 수동 스냅샷WordPress-1으로 복사us-west-2합니다us-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
```

출력:

```
{
    "operations": [
        {
            "id": "dffa128b-0b07-476e-b390-bEXAMPLE3775",
            "resourceName": "WordPress-1-10142019",
            "resourceType": "InstanceSnapshot",
            "createdAt": 1571082493.422,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-east-1"
```

```
            },
            "isTerminal": false,
            "operationDetails": "us-west-2:WordPress-1",
            "operationType": "CopySnapshot",
            "status": "Started",
            "statusChangedAt": 1571082493.422
        }
    ]
}
```

자세한 내용은 Lightsail Dev Guide의 Amazon Lightsail에서 인스턴스 또는 디스크의 자동 스냅샷 유지를 참조하세요.

- 자세한 API 내용은 명령 참조CopySnapshot의 섹션을 참조하세요. AWS CLI

### `create-disk-from-snapshot`

다음 코드 예시에서는 `create-disk-from-snapshot`을 사용하는 방법을 보여 줍니다.

AWS CLI

디스크 스냅샷에서 디스크를 생성하려면

다음 `create-disk-from-snapshot` 예제에서는 지정된 블록 스토리지 디스크 스냅샷`Disk-2`에서 라는 블록 스토리지 디스크를 생성합니다. 디스크는 32GB의 스토리지 공간이 있는 지정된 AWS 리전 및 가용 영역에 생성됩니다.

```
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
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 Lightsail 개발자 안내서의 Amazon Lightsail의 스냅샷에서 블록 스토리지 디스크 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateDiskFromSnapshot의 섹션을 참조하세요. AWS CLI

### `create-disk-snapshot`

다음 코드 예시에서는 `create-disk-snapshot`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 디스크 스냅샷 생성

다음 `create-disk-snapshot` 예제에서는 지정된 블록 스토리지 디스크DiskSnapshot-1의 라는 스냅샷을 생성합니다.

```
aws lightsail create-disk-snapshot \
    --disk-name Disk-1 \
    --disk-snapshot-name DiskSnapshot-1
```

출력:

```
{
    "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
        }
    ]
}
```

예제 2: 인스턴스의 시스템 디스크 스냅샷 생성

다음 `create-disk-snapshot` 예제에서는 지정된 인스턴스의 시스템 디스크의 스냅샷을 생성합니다.

```
aws lightsail create-disk-snapshot \
    --instance-name WordPress-1 \
    --disk-snapshot-name SystemDiskSnapshot-1
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 Lightsail 개발자 안내서의 Amazon Lightsail 스냅샷 및 Amazon Lightsail에서 인스턴스 루트 볼륨의 스냅샷 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateDiskSnapshot의 섹션을 참조하세요. AWS CLI

### `create-disk`

다음 코드 예시에서는 `create-disk`을 사용하는 방법을 보여 줍니다.

AWS CLI

블록 스토리지 디스크를 생성하려면

다음 create-disk 예제에서는 32GB의 스토리지 공간이 있는 지정된 AWS 리전 및 가용 영역에 블록 스토리지 디스크Disk-1를 생성합니다.

```
aws lightsail create-disk \
    --disk-name Disk-1 \
    --availability-zone us-west-2a \
    --size-in-gb 32
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조CreateDisk의 섹션을 참조하세요. AWS CLI

### create-domain-entry

다음 코드 예시에서는 create-domain-entry을 사용하는 방법을 보여 줍니다.

AWS CLI

도메인 항목을 생성하려면(DNS 레코드)

다음 create-domain-entry 예제에서는 인스턴스의 IP 주소를 가리키는 지정된 도메인의 정점에 대한 DNS 레코드(A)를 생성합니다.

참고: Lightsail의 도메인 관련 API 작업은 us-east-1 리전에서만 사용할 수 있습니다. CLI 프로파일이 다른 리전을 사용하도록 구성된 경우 --region us-east-1 파라미터를 포함해야 합니다. 그렇지 않으면 명령이 실패합니다.

```
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
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 Amazon DNS Amazon Lightsail 개발자 안내서의 Amazon Lightsail에서 를 참조하고, Lightsail 개발자 안내서의 Amazon Lightsail에서 도메인 DNS 레코드를 관리할 DNS 영역 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateDomainEntry의 섹션을 참조하세요. AWS CLI

### create-domain

다음 코드 예시에서는 create-domain을 사용하는 방법을 보여 줍니다.

AWS CLI

도메인(DNS 영역)을 생성하려면

다음 create-domain 예제에서는 지정된 도메인에 대한 DNS 영역을 생성합니다.

참고: Lightsail의 도메인 관련 API 작업은 us-east-1 리전에서만 사용할 수 있습니다. CLI 프로파일이 다른 리전을 사용하도록 구성된 경우 --region us-east-1 파라미터를 포함해야 합니다. 그렇지 않으면 명령이 실패합니다.

```
aws lightsail create-domain \
    --region us-east-1 \
    --domain-name example.com
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 Amazon DNS Amazon Lightsail 개발자 안내서의 Amazon Lightsail에서 를 참조하고, Lightsail 개발자 안내서의 Amazon Lightsail에서 도메인 DNS 레코드를 관리할 DNS 영역 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateDomain의 섹션을 참조하세요. AWS CLI

### create-instance-snapshot

다음 코드 예시에서는 create-instance-snapshot을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스의 스냅샷을 생성하려면

다음 create-instance-snapshot 예제에서는 지정된 인스턴스에서 스냅샷을 생성합니다.

```
aws lightsail create-instance-snapshot \
    --instance-name WordPress-1 \
    --instance-snapshot-name WordPress-Snapshot-1
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조CreateInstanceSnapshot의 섹션을 참조하세요. AWS CLI

### `create-instances-from-snapshot`

다음 코드 예시에서는 `create-instances-from-snapshot`을 사용하는 방법을 보여 줍니다.

AWS CLI

스냅샷에서 인스턴스를 생성하려면

다음 `create-instances-from-snapshot` 예제에서는 $12 USD 번들을 사용하여 지정된 AWS 리전 및 가용 영역에 지정된 인스턴스 스냅샷에서 인스턴스를 생성합니다.

참고: 지정한 번들은 스냅샷을 생성하는 데 사용된 원본 소스 인스턴스의 번들과 사양이 같거나 커야 합니다.

```
aws lightsail create-instances-from-snapshot \
    --instance-snapshot-name WordPress-1-1569866208 \
    --instance-names WordPress-2 \
    --availability-zone us-west-2a \
    --bundle-id small_3_0
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조CreateInstancesFromSnapshot의 섹션을 참조하세요. AWS CLI

### `create-instances`

다음 코드 예시에서는 `create-instances`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 단일 인스턴스 생성

다음 `create-instances` 예제에서는 WordPress 블루프린트와 $5.00 USD 번들을 사용하여 지정된 AWS 리전 및 가용 영역에 인스턴스를 생성합니다.

```
aws lightsail create-instances \
    --instance-names Instance-1 \
    --availability-zone us-west-2a \
    --blueprint-id wordpress \
    --bundle-id nano_3_0
```

출력:

```
{
    "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
        }
    ]
}
```

예제 2: 한 번에 여러 인스턴스 생성

다음 `create-instances` 예제에서는 WordPress 블루프린트와 $5.00 USD 번들을 사용하여 지정된 AWS 리전 및 가용 영역에 인스턴스 3개를 생성합니다.

```
aws lightsail create-instances \
    --instance-names {"Instance1","Instance2","Instance3"} \
    --availability-zone us-west-2a \
    --blueprint-id wordpress \
    --bundle-id nano_3_0
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조CreateInstances의 섹션을 참조하세요. AWS CLI

### create-key-pair

다음 코드 예시에서는 create-key-pair을 사용하는 방법을 보여 줍니다.

AWS CLI

키 페어를 생성하는 방법

다음 create-key-pair 예제에서는 인스턴스를 인증하고 연결하는 데 사용할 수 있는 키 페어를 생성합니다.

```
aws lightsail create-key-pair \
    --key-pair-name MyPersonalKeyPair
```

출력은 생성된 키 페어를 사용하는 인스턴스에 인증하는 데 사용할 수 있는 프라이빗 키 base64 값을 제공합니다. 참고: 나중에 검색할 수 없으므로 프라이빗 키 base64 값을 복사하여 안전한 위치에 붙여넣습니다.

```
{
    "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
 AAAAB3NzaC1yc2EAAAADAQABAAABAQCV0xUEwx96amPERH7K1bVT1tTFl9OmNk6o7m5YVHk9xlOdMbDRbFvhtXvw4jz
+BHUgedGUXno6uF7agqxZNO1kPLJBIVTW26SSYBJ0tE
+y804UyVsjrbUqCaMXDhmfXpWulMPwuXhwcKh7e8hwoTfkiX0E6Ql
+KqF/MiA3w6DCjEqvvdIO7SiEZJFsuGNfYDDN3w60Rel5MUhmn3OJdn4y/
A7NWb3IxL4pPfVE4rgFRKU8n1jp9kwRnlVMVBOWuGXk6n+H6M2f1 ",
    "privateKeyBase64": "-----BEGIN RSA PRIVATE KEY-----
EXAMPLETCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
\nVVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6\nb24xFDASBgNVBAsTC0lBTSBD
\nBgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
\nMTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
\nVQQHEwdTZWF0dGxlMQ8wDQEXAMPLEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
\nb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
\nYXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMEXAMPLE4GmWIWJ
\n21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
\nrDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
\nIbb3OhjZnzcvQAaREXAMPLEMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4\nnUhVVxYUntneD9+h8Mg9q6q
+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
\nFFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OEXAMPLELvjx79LjSTb
\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
    }
}
```


- 자세한 API 내용은 명령 참조CreateKeyPair의 섹션을 참조하세요. AWS CLI

### create-load-balancer-tls-certificate

다음 코드 예시에서는 create-load-balancer-tls-certificate을 사용하는 방법을 보여 줍니다.

AWS CLI

로드 밸런서에 대한 TLS 인증서를 생성하려면

다음 create-load-balancer-tls-certificate 예제에서는 지정된 로드 밸런서에 연결된 TLS 인증서를 생성합니다. 생성된 인증서는 지정된 도메인에 적용됩니다. 참고: 로드 밸런서에 대해 두 개의 인증서만 생성할 수 있습니다.

```
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
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조CreateLoadBalancerTlsCertificate의 섹션을 참조하세요. AWS CLI

### `create-load-balancer`

다음 코드 예시에서는 `create-load-balancer`을 사용하는 방법을 보여 줍니다.

AWS CLI

로드 밸런서를 생성하려면

다음 `create-load-balancer` 예제에서는 TLS 인증서가 있는 로드 밸런서를 생성합니다. TLS 인증서는 지정된 도메인에 적용되며 트래픽을 포트 80의 인스턴스로 라우팅합니다.

```
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
```

출력:

```
{
    "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
        }
    ]
}
```


자세한 내용은 Lightsail 개발자 안내서의 Lightsail 로드 밸런서를 참조하세요.

- 자세한 API 내용은 명령 참조CreateLoadBalancer의 섹션을 참조하세요. AWS CLI

### create-relational-database-from-snapshot

다음 코드 예시에서는 create-relational-database-from-snapshot을 사용하는 방법을 보여 줍니다.

AWS CLI

스냅샷에서 관리형 데이터베이스를 생성하려면

다음 create-relational-database-from-snapshot 예제에서는 $15 USD 표준 데이터베이스 번들을 사용하여 지정된 AWS 리전 및 가용 영역의 지정된 스냅샷에서 관리형 데이터베이스를 생성합니다. 참고: 지정한 번들은 스냅샷을 생성하는 데 사용된 원본 소스 데이터베이스의 번들과 사양이 같거나 커야 합니다.

```
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
```

출력:

```
{
    "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
        }
```

```
    ]
}
```

- 자세한 API 내용은 명령 참조CreateRelationalDatabaseFromSnapshot의 섹션을 참조하세요. AWS CLI

### `create-relational-database-snapshot`

다음 코드 예시에서는 `create-relational-database-snapshot`을 사용하는 방법을 보여 줍니다.

AWS CLI

관리형 데이터베이스의 스냅샷을 생성하려면

다음 `create-relational-database-snapshot` 예제에서는 지정된 관리형 데이터베이스의 스냅샷을 생성합니다.

```
aws lightsail create-relational-database-snapshot \
    --relational-database-name Database1 \
    --relational-database-snapshot-name RelationalDatabaseSnapshot1
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조CreateRelationalDatabaseSnapshot의 섹션을 참조하세요. AWS CLI

### `create-relational-database`

다음 코드 예시에서는 `create-relational-database`을 사용하는 방법을 보여 줍니다.

AWS CLI

관리형 데이터베이스를 생성하려면

다음 `create-relational-database` 예제에서는 MySQL 5.6 데이터베이스 엔진(mysql_5_6)과 $15 USD 표준 데이터베이스 번들(micro_1_0)을 사용하여 지정된 AWS 리전 및 가용 영역에 관리형 데이터베이스를 생성합니다. 관리형 데이터베이스는 마스터 사용자 이름으로 미리 채워지며 공개적으로 액세스할 수 없습니다.

```
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
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조CreateRelationalDatabase의 섹션을 참조하세요. AWS CLI

### delete-auto-snapshot

다음 코드 예시에서는 delete-auto-snapshot을 사용하는 방법을 보여 줍니다.

AWS CLI

자동 스냅샷을 삭제하려면

다음 delete-auto-snapshot 예제에서는 인스턴스 2019-10-10의 자동 스냅샷을 삭제합니다WordPress-1.

```
aws lightsail delete-auto-snapshot \
    --resource-name WordPress-1 \
    --date 2019-10-10
```

출력:

```
{
```

```
    "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"
        }
    ]
}
```

자세한 내용은 Lightsail 개발 가이드의 Amazon Lightsail에서 인스턴스 또는 디스크의 자동 스냅샷 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteAutoSnapshot의 섹션을 참조하세요. AWS CLI

### `delete-disk-snapshot`

다음 코드 예시에서는 `delete-disk-snapshot`을 사용하는 방법을 보여 줍니다.

AWS CLI

블록 스토리지 디스크의 스냅샷을 삭제하려면

다음 `delete-disk-snapshot` 예제에서는 블록 스토리지 디스크의 지정된 스냅샷을 삭제합니다.

```
aws lightsail delete-disk-snapshot \
    --disk-snapshot-name DiskSnapshot-1
```

출력:

```
{
    "operations": [
        {
            "id": "d1e5766d-b81e-4595-ad5d-02afbccfcd5d",
```

```
            "resourceName": "DiskSnapshot-1",
            "resourceType": "DiskSnapshot",
            "createdAt": 1569873552.79,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationType": "DeleteDiskSnapshot",
            "status": "Succeeded",
            "statusChangedAt": 1569873552.79
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DeleteDiskSnapshot의 섹션을 참조하세요. AWS CLI

### delete-disk

다음 코드 예시에서는 delete-disk을 사용하는 방법을 보여 줍니다.

AWS CLI

블록 스토리지 디스크를 삭제하려면

다음 delete-disk 예제에서는 지정된 블록 스토리지 디스크를 삭제합니다.

```
aws lightsail delete-disk \
    --disk-name Disk-1
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DeleteDisk의 섹션을 참조하세요. AWS CLI

### delete-domain-entry

다음 코드 예시에서는 delete-domain-entry을 사용하는 방법을 보여 줍니다.

AWS CLI

도메인 항목을 삭제하려면(DNS 레코드)

다음 delete-domain-entry 예제에서는 기존 도메인에서 지정된 도메인 항목을 삭제합니다.

참고: Lightsail의 도메인 관련 API 작업은 us-east-1 리전에서만 사용할 수 있습니다. CLI 프로파일이 다른 리전을 사용하도록 구성된 경우 --region us-east-1 파라미터를 포함해야 합니다. 그렇지 않으면 명령이 실패합니다.

```
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
```

출력:

```
{
    "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
    }
}
```

- 자세한 API 내용은 명령 참조DeleteDomainEntry의 섹션을 참조하세요. AWS CLI

### `delete-domain`

다음 코드 예시에서는 `delete-domain`을 사용하는 방법을 보여 줍니다.

AWS CLI

도메인(DNS 영역)을 삭제하려면

다음 `delete-domain` 예제에서는 지정된 도메인과 도메인( 레코드)DNS의 모든 항목을 삭제합니다.

참고: Lightsail의 도메인 관련 API 작업은 `us-east-1` 리전에서만 사용할 수 있습니다. CLI 프로파일이 다른 리전을 사용하도록 구성된 경우 `--region us-east-1` 파라미터를 포함해야 합니다. 그렇지 않으면 명령이 실패합니다.

```
aws lightsail delete-domain \
    --region us-east-1 \
    --domain-name example.com
```

출력:

```
{
    "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
    }
}
```

- 자세한 API 내용은 명령 참조DeleteDomain의 섹션을 참조하세요. AWS CLI

### delete-instance-snapshot

다음 코드 예시에서는 delete-instance-snapshot을 사용하는 방법을 보여 줍니다.

AWS CLI

title

다음 delete-instance-snapshot 예제에서는 인스턴스의 지정된 스냅샷을 삭제합니다.

```
aws lightsail delete-instance-snapshot \
    --instance-snapshot-name WordPress-1-Snapshot-1
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DeleteInstanceSnapshot의 섹션을 참조하세요. AWS CLI

### delete-instance

다음 코드 예시에서는 delete-instance을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스를 삭제하려면

다음 delete-instance 예제에서는 지정된 인스턴스를 삭제합니다.

```
aws lightsail delete-instance \
    --instance-name WordPress-1
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DeleteInstance의 섹션을 참조하세요. AWS CLI

### `delete-key-pair`

다음 코드 예시에서는 `delete-key-pair`을 사용하는 방법을 보여 줍니다.

AWS CLI

키 페어를 삭제하는 방법

다음 `delete-key-pair` 예제에서는 지정된 키 페어를 삭제합니다.

```
aws lightsail delete-key-pair \
    --key-pair-name MyPersonalKeyPair
```

출력:

```
{
```

```
    "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
    }
}
```

- 자세한 API 내용은 명령 참조DeleteKeyPair의 섹션을 참조하세요. AWS CLI

### delete-known-host-keys

다음 코드 예시에서는 delete-known-host-keys을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스에서 알려진 호스트 키를 삭제하려면

다음 delete-known-host-keys 예제에서는 지정된 인스턴스에서 알려진 호스트 키를 삭제합니다.

```
aws lightsail delete-known-host-keys \
    --instance-name Instance-1
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 Lightsail 개발 가이드의 Amazon Lightsail 브라우저 기반 SSH 또는 RDP 클라이언트와의 연결 문제 해결을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteKnownHostKeys의 섹션을 참조하세요. AWS CLI

### `delete-load-balancer-tls-certificate`

다음 코드 예시에서는 `delete-load-balancer-tls-certificate`을 사용하는 방법을 보여 줍니다.

AWS CLI

로드 밸런서의 TLS 인증서를 삭제하려면

다음 `delete-load-balancer-tls-certificate` 예제에서는 지정된 로드 밸런서에서 지정 TLS 인증서를 삭제합니다.

```
aws lightsail delete-load-balancer-tls-certificate \
    --load-balancer-name MyFirstLoadBalancer \
    --certificate-name MyFirstCertificate
```

출력:

```
{
    "operations": [
        {
            "id": "50bec274-e45e-4caa-8a69-b763ef636583",
            "resourceName": "MyFirstCertificate",
            "resourceType": "LoadBalancerTlsCertificate",
```

```
            "createdAt": 1569874989.48,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationType": "DeleteLoadBalancerTlsCertificate",
            "status": "Started",
            "statusChangedAt": 1569874989.48
        },
        {
            "id": "78c58cdc-a59a-4b27-8213-500638634a8f",
            "resourceName": "MyFirstLoadBalancer",
            "resourceType": "LoadBalancer",
            "createdAt": 1569874989.48,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": false,
            "operationType": "DeleteLoadBalancerTlsCertificate",
            "status": "Started",
            "statusChangedAt": 1569874989.48
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DeleteLoadBalancerTlsCertificate의 섹션을 참조하세요. AWS CLI

### `delete-load-balancer`

다음 코드 예시에서는 `delete-load-balancer`을 사용하는 방법을 보여 줍니다.

AWS CLI

로드 밸런서를 삭제하는 방법

다음 `delete-load-balancer` 예제에서는 지정된 로드 밸런서와 관련 TLS 인증서를 삭제합니다.

```
aws lightsail delete-load-balancer \
    --load-balancer-name MyFirstLoadBalancer
```

출력:


```
{
    "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
        }
    ]
}
```

자세한 내용은 가이드의 제목을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteLoadBalancer의 섹션을 참조하세요. AWS CLI

### delete-relational-database-snapshot

다음 코드 예시에서는 delete-relational-database-snapshot을 사용하는 방법을 보여 줍니다.

AWS CLI

관리형 데이터베이스의 스냅샷을 삭제하려면

다음 delete-relational-database-snapshot 예제에서는 관리형 데이터베이스의 지정된 스냅샷을 삭제합니다.

```
aws lightsail delete-relational-database-snapshot \
    --relational-database-snapshot-name Database-Oregon-1-1566839359
```

출력:

```
{
    "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
        }
```

```
    ]
}
```

- 자세한 API 내용은 명령 참조DeleteRelationalDatabaseSnapshot의 섹션을 참조하세요. AWS CLI

### delete-relational-database

다음 코드 예시에서는 delete-relational-database을 사용하는 방법을 보여 줍니다.

AWS CLI

관리형 데이터베이스를 삭제하려면

다음 delete-relational-database 예제에서는 지정된 관리형 데이터베이스를 삭제합니다.

```
aws lightsail delete-relational-database \
    --relational-database-name Database-1
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DeleteRelationalDatabase의 섹션을 참조하세요. AWS CLI

### `detach-static-ip`

다음 코드 예시에서는 `detach-static-ip`을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스에서 정적 IP를 분리하려면

다음 `detach-static-ip` 예제에서는 연결된 인스턴스StaticIp-1에서 정적 IP를 분리합니다.

```
aws lightsail detach-static-ip \
    --static-ip-name StaticIp-1
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DetachStaticIp의 섹션을 참조하세요. AWS CLI

### `get-active-names`

다음 코드 예시에서는 `get-active-names`을 사용하는 방법을 보여 줍니다.

AWS CLI

활성 리소스 이름을 가져오려면

다음 `get-active-names` 예제에서는 구성된 AWS 리전의 활성 리소스 이름을 반환합니다.

```
aws lightsail get-active-names
```

출력:

```
{
    "activeNames": [
        "WordPress-1",
        "StaticIp-1",
        "MEAN-1",
        "Plesk_Hosting_Stack_on_Ubuntu-1"
    ]
}
```

- 자세한 API 내용은 명령 참조GetActiveNames의 섹션을 참조하세요. AWS CLI

### `get-auto-snapshots`

다음 코드 예시에서는 `get-auto-snapshots`을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스에 사용할 수 있는 자동 스냅샷을 가져오려면

다음 `get-auto-snapshots` 예제에서는 인스턴스 에 사용 가능한 자동 스냅샷을 반환합니다WordPress-1.

```
aws lightsail get-auto-snapshots \
    --resource-name WordPress-1
```

출력:

```
{
    "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": []
        }
    ]
}
```

자세한 내용은 Lightsail Dev Guide의 Amazon Lightsail에서 인스턴스 또는 디스크의 자동 스냅샷 유지를 참조하세요.

- 자세한 API 내용은 명령 참조GetAutoSnapshots의 섹션을 참조하세요. AWS CLI

### `get-blueprints`

다음 코드 예시에서는 `get-blueprints`을 사용하는 방법을 보여 줍니다.

AWS CLI

새 인스턴스의 청사진을 가져오려면

다음 `get-blueprints` 예제에서는 Amazon Lightsail 에서 새 인스턴스를 생성하는 데 사용할 수 있는 사용 가능한 모든 청사진에 대한 세부 정보를 보여줍니다.

```
aws lightsail get-blueprints
```

출력:

```
{
    "blueprints": [
        {
            "blueprintId": "wordpress",
            "name": "WordPress",
            "group": "wordpress",
            "type": "app",
            "description": "Bitnami, the leaders in application packaging, and
 Automattic, the experts behind WordPress, have teamed up to offer this official
 WordPress image. This image is a pre-configured, ready-to-run image for running
 WordPress on Amazon Lightsail. WordPress is the world's most popular content
 management platform. Whether it's for an enterprise or small business website, or
 a personal or corporate blog, content authors can easily create content using its
 new Gutenberg editor, and developers can extend the base platform with additional
 features. Popular plugins like Jetpack, Akismet, All in One SEO Pack, WP Mail,
 Google Analytics for WordPress, and Amazon Polly are all pre-installed in this
 image. Let's Encrypt SSL certificates are supported through an auto-configuration
 script.",
            "isActive": true,
            "minPower": 0,
            "version": "6.5.3-0",
            "versionCode": "1",
            "productUrl": "https://aws.amazon.com/marketplace/pp/B00NN8Y43U",
            "licenseUrl": "https://aws.amazon.com/marketplace/pp/B00NN8Y43U#pdp-
usage",
            "platform": "LINUX_UNIX"
        },
        {
            "blueprintId": "lamp_8_bitnami",
            "name": "LAMP (PHP 8)",
            "group": "lamp_8",
            "type": "app",
            "description": "LAMP with PHP 8.X packaged by Bitnami enables you
 to quickly start building your websites and applications by providing a coding
 framework. As a developer, it provides standalone project directories to store your
 applications. This blueprint is configured for production environments. It includes
 SSL auto-configuration with Let's Encrypt certificates, and the latest releases of
 PHP, Apache, and MariaDB on Linux. This application also includes phpMyAdmin, PHP
 main modules and Composer.",
```

```
            "isActive": true,
            "minPower": 0,
            "version": "8.2.18-4",
            "versionCode": "1",
            "productUrl": "https://aws.amazon.com/marketplace/pp/
prodview-6g3gzfcih6dvu",
            "licenseUrl": "https://aws.amazon.com/marketplace/pp/
prodview-6g3gzfcih6dvu#pdp-usage",
            "platform": "LINUX_UNIX"
        },
        {
            "blueprintId": "nodejs",
            "name": "Node.js",
            "group": "node",
            "type": "app",
            "description": "Node.js packaged by Bitnami is a pre-configured, ready
 to run image for Node.js on Amazon EC2. It includes the latest version of Node.js,
 Apache, Python and Redis. The image supports multiple Node.js applications, each
 with its own virtual host and project directory. It is configured for production
 use and is secure by default, as all ports except HTTP, HTTPS and SSH ports are
 closed. Let's Encrypt SSL certificates are supported through an auto-configuration
 script. Developers benefit from instant access to a secure, update and consistent
 Node.js environment without having to manually install and configure multiple
 components and libraries.",
            "isActive": true,
            "minPower": 0,
            "version": "18.20.2-0",
            "versionCode": "1",
            "productUrl": "https://aws.amazon.com/marketplace/pp/B00NNZUAKO",
            "licenseUrl": "https://aws.amazon.com/marketplace/pp/B00NNZUAKO#pdp-
usage",
            "platform": "LINUX_UNIX"
        },
        ...
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetBlueprints의 섹션을 참조하세요. AWS CLI

### get-bundles

다음 코드 예시에서는 get-bundles을 사용하는 방법을 보여 줍니다.

AWS CLI

새 인스턴스에 대한 번들을 가져오려면

다음 `get-bundles` 예제에서는 Amazon Lightsail 에서 새 인스턴스를 생성하는 데 사용할 수 있는 사용 가능한 모든 번들에 대한 세부 정보를 표시합니다.

```
aws lightsail get-bundles
```

출력:

```
{
    "bundles": [
        {
            "price": 5.0,
            "cpuCount": 2,
            "diskSizeInGb": 20,
            "bundleId": "nano_3_0",
            "instanceType": "nano",
            "isActive": true,
            "name": "Nano",
            "power": 298,
            "ramSizeInGb": 0.5,
            "transferPerMonthInGb": 1024,
            "supportedPlatforms": [
                "LINUX_UNIX"
            ]
        },
        {
            "price": 7.0,
            "cpuCount": 2,
            "diskSizeInGb": 40,
            "bundleId": "micro_3_0",
            "instanceType": "micro",
            "isActive": true,
            "name": "Micro",
            "power": 500,
            "ramSizeInGb": 1.0,
            "transferPerMonthInGb": 2048,
            "supportedPlatforms": [
                "LINUX_UNIX"
            ]
        },
```

```
        {
            "price": 12.0,
            "cpuCount": 2,
            "diskSizeInGb": 60,
            "bundleId": "small_3_0",
            "instanceType": "small",
            "isActive": true,
            "name": "Small",
            "power": 1000,
            "ramSizeInGb": 2.0,
            "transferPerMonthInGb": 3072,
            "supportedPlatforms": [
                "LINUX_UNIX"
            ]
        },
        ...
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetBundles의 섹션을 참조하세요. AWS CLI

### `get-cloud-formation-stack-records`

다음 코드 예시에서는 `get-cloud-formation-stack-records`을 사용하는 방법을 보여 줍니다.

AWS CLI

CloudFormation 스택 레코드 및 관련 스택을 가져오려면

다음 `get-cloud-formation-stack-records` 예제에서는 내보낸 Amazon Lightsail 스냅샷에서 Amazon EC2 리소스를 생성하는 데 사용되는 CloudFormation 스택 레코드 및 관련 스택에 대한 세부 정보를 표시합니다.

```
aws lightsail get-cloud-formation-stack-records
```

출력:

```
{
    "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"
            }
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetCloudFormationStackRecords의 섹션을 참조하세요. AWS CLI

### `get-disk-snapshot`

다음 코드 예시에서는 `get-disk-snapshot`을 사용하는 방법을 보여 줍니다.

AWS CLI

디스크 스냅샷에 대한 정보를 가져오려면

다음 `get-disk-snapshot` 예제에서는 디스크 스냅샷 에 대한 세부 정보를 표시합니다`Disk-1-1566839161`.

```
aws lightsail get-disk-snapshot \
    --disk-snapshot-name Disk-1-1566839161
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 가이드의 제목을 참조하세요.

- 자세한 API 내용은 명령 참조GetDiskSnapshot의 섹션을 참조하세요. AWS CLI

### get-disk-snapshots

다음 코드 예시에서는 get-disk-snapshots을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 디스크 스냅샷에 대한 정보를 가져오려면

다음 get-disk-snapshots 예제에서는 구성된 AWS 리전의 모든 디스크 스냅샷에 대한 세부 정보를 표시합니다.

```
aws lightsail get-disk-snapshots
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetDiskSnapshots의 섹션을 참조하세요. AWS CLI

### get-disk

다음 코드 예시에서는 get-disk을 사용하는 방법을 보여 줍니다.

AWS CLI

블록 스토리지 디스크에 대한 정보를 가져오려면

다음 get-disk 예제에서는 디스크 에 대한 세부 정보를 표시합니다Disk-1.

```
aws lightsail get-disk \
    --disk-name Disk-1
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 가이드의 제목을 참조하세요.

- 자세한 API 내용은 명령 참조GetDisk의 섹션을 참조하세요. AWS CLI

### get-disks

다음 코드 예시에서는 get-disks을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 블록 스토리지 디스크에 대한 정보를 가져오려면

다음 get-disks 예제에서는 구성된 AWS 리전의 모든 디스크에 대한 세부 정보를 표시합니다.

```
aws lightsail get-disks
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetDisks의 섹션을 참조하세요. AWS CLI

### `get-domain`

다음 코드 예시에서는 `get-domain`을 사용하는 방법을 보여 줍니다.

AWS CLI

도메인에 대한 정보를 가져오려면

다음 `get-domain` 예제에서는 도메인 에 대한 세부 정보를 표시합니다`example.com`.

참고: Lightsail의 도메인 관련 API 작업은 `us-east-1` AWS 리전에서만 사용할 수 있습니다. CLI 프로파일이 다른 리전을 사용하도록 구성된 경우 `` --region us-east-1`` 파라미터를 포함해야 합니다. 그렇지 않으면 명령이 실패합니다.

```
aws lightsail get-domain \
    --domain-name example.com \
    --region us-east-1
```

출력:

```
{
    "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"
            }
        ]
    }
}
```

- 자세한 API 내용은 명령 참조GetDomain의 섹션을 참조하세요. AWS CLI

### get-domains

다음 코드 예시에서는 get-domains을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 도메인에 대한 정보를 가져오려면

다음 get-domains 예제에서는 구성된 AWS 리전의 모든 도메인에 대한 세부 정보를 표시합니다.

참고: Lightsail의 도메인 관련 API 작업은 us-east-1 AWS 리전에서만 사용할 수 있습니다. CLI 프로파일이 다른 리전을 사용하도록 구성된 경우 --region us-east-1 파라미터를 포함해야 합니다. 그렇지 않으면 명령이 실패합니다.

```
aws lightsail get-domains \
    --region us-east-1
```

출력:


```
{
    "domains": [
        {
            "name": "example.com",
            "arn":
 "arn:aws:lightsail:global:111122223333:Domain/28cda903-3f15-44b2-9baf-3EXAMPLEb304",
            "supportCode": "6EXAMPLE3362//hostedzone/ZEXAMPLEONGSC1",
            "createdAt": 1570728588.6,
            "location": {
                "availabilityZone": "all",
                "regionName": "global"
            },
            "resourceType": "Domain",
            "tags": [],
            "domainEntries": [
                {
                    "id": "-1682899164",
                    "name": "example.com",
                    "target": "192.0.2.0",
                    "isAlias": false,
                    "type": "A"
                },
                {
                    "id": "1703104243",
                    "name": "example.com",
                    "target": "ns-137.awsdns-17.com",
                    "isAlias": false,
                    "type": "NS"
                },
                {
                    "id": "-1038331153",
                    "name": "example.com",
                    "target": "ns-4567.awsdns-21.co.uk",
                    "isAlias": false,
                    "type": "NS"
                },
                {
                    "id": "-2107289565",
                    "name": "example.com",
                    "target": "ns-333.awsdns-22.net",
                    "isAlias": false,
                    "type": "NS"
```

```
                },
                {
                    "id": "1582095705",
                    "name": "example.com",
                    "target": "ns-1111.awsdns-51.org",
                    "isAlias": false,
                    "type": "NS"
                },
                {
                    "id": "-1769796132",
                    "name": "example.com",
                    "target": "ns-1234.awsdns-21.co.uk. awsdns-
hostmaster.amazon.com. 1 7200 900 1209600 86400",
                    "isAlias": false,
                    "type": "SOA"
                },
                {
                    "id": "1029454894",
                    "name": "_dead6a124ede046a0319eb44a4eb3cbc.example.com",
                    "target": "_be133b0a0899fb7b6bf79d9741d1a383.hkvuiqjoua.acm-
validations.aws",
                    "isAlias": false,
                    "type": "CNAME"
                }
            ]
        },
        {
            "name": "example.net",
            "arn": "arn:aws:lightsail:global:111122223333:Domain/9c9f0d70-
c92e-4753-86c2-6EXAMPLE029d",
            "supportCode": "6EXAMPLE3362//hostedzone/ZEXAMPLE5TPKMV",
            "createdAt": 1556661071.384,
            "location": {
                "availabilityZone": "all",
                "regionName": "global"
            },
            "resourceType": "Domain",
            "tags": [],
            "domainEntries": [
                {
                    "id": "-766320943",
                    "name": "example.net",
                    "target": "192.0.2.2",
                    "isAlias": false,
```

```
                    "type": "A"
                },
                {
                    "id": "-453913825",
                    "name": "example.net",
                    "target": "ns-123.awsdns-10.net",
                    "isAlias": false,
                    "type": "NS"
                },
                {
                    "id": "1553601564",
                    "name": "example.net",
                    "target": "ns-4444.awsdns-47.co.uk",
                    "isAlias": false,
                    "type": "NS"
                },
                {
                    "id": "1653797661",
                    "name": "example.net",
                    "target": "ns-7890.awsdns-61.org",
                    "isAlias": false,
                    "type": "NS"
                },
                {
                    "id": "706414698",
                    "name": "example.net",
                    "target": "ns-123.awsdns-44.com",
                    "isAlias": false,
                    "type": "NS"
                },
                {
                    "id": "337271745",
                    "name": "example.net",
                    "target": "ns-4444.awsdns-47.co.uk. awsdns-
hostmaster.amazon.com. 1 7200 900 1209600 86400",
                    "isAlias": false,
                    "type": "SOA"
                },
                {
                    "id": "-1785431096",
                    "name": "www.example.net",
                    "target": "192.0.2.2",
                    "isAlias": false,
                    "type": "A"
```

```
                }
            ]
        },
        {
            "name": "example.org",
            "arn": "arn:aws:lightsail:global:111122223333:Domain/
f0f13ba3-3df0-4fdc-8ebb-1EXAMPLEf26e",
            "supportCode": "6EXAMPLE3362//hostedzone/ZEXAMPLEAFO38",
            "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"
                }
            ]
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetDomains의 섹션을 참조하세요. AWS CLI

### get-export-snapshot-record

다음 코드 예시에서는 get-export-snapshot-record을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon으로 내보낸 스냅샷 레코드를 가져오려면 EC2

다음 get-export-snapshot-record 예제에서는 Amazon 로 내보낸 Amazon Lightsail 인스턴스 또는 디스크 스냅샷에 대한 세부 정보를 표시합니다EC2.

```
aws lightsail get-export-snapshot-records
```

출력:

```
{
    "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-Oregon-1-1540833565",
                "arn": "arn:aws:lightsail:us-
west-2:111122223333:InstanceSnapshot/82334399-b5f2-49ec-8382-0EXAMPLEe45f",
                "fromResourceName": "LAMP_PHP_5-512MB-Oregon-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"
            }
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetExportSnapshotRecord의 섹션을 참조하세요. AWS CLI

### get-instance-access-details

다음 코드 예시에서는 get-instance-access-details을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스의 호스트 키 정보를 가져오려면

다음 `get-instance-access-details` 예제에서는 인스턴스 에 대한 호스트 키 정보를 보여줍니다`WordPress_Multisite-1`.

```
aws lightsail get-instance-access-details \
    --instance-name WordPress_Multisite-1
```

출력:

```
{
    "accessDetails": {
        "certKey": "ssh-rsa-cert-v01@openssh.com
 AEXAMPLEaC1yc2EtY2VydC12MDFAb3BlbnNzaC5jb20AAAAgNf076Dt3ppmPd0fPxZVMmS491aEAYYH9cHqAJ3fNML8
vEXAMPLE2eBWJyQvn7ol/
i0+s966h5sx8qUD79lPB7q5UESd5VZGFtytrykfQJnjiwqe7EV5agzvjblLj26Fb37EKda9HVfCOu8pWbvky7Tyn9w29
+xMfQM9xVz0rXZmqx8uJidJpRgLCMTviofwQJU/
K1EXAMPLEAAAAAAAABAAAALS00MzMzMDU4MzA4ODg1MTY2NjM4Onp6UWlndHk4UElRSG9STitOTG5QSEE9PQAAAAsAAA
+LiB+ozNbUA0cdNL9Y67x7qPv/R7XhTc21+2A+8+GuVpK/Kz9dqDMKNAEXAMPLE+YYN
+tiXm7Y8OgziK+7iDB7xUuQ4vghmn4+qgz9mKwYgWvVe2+0XLuV7cnWPB7iUlHQg
+E3LUKrV4ZFw9pj7X2dFdNKfMxwWgI1ISWKimEXAMPLEeHjrf1Rqc/
QH6TpWCvPfcx8uvwVqdwTfkE/SfA5BCzbGGI1UmIUadh8nHcb5FamQ1hK7kECy47K/x9FMn/
KwmM7pCwJbSLDMO7n9bnbvck6m8ZoB2N2YLMG5dW7BerEXAMPLEobqfdtyYJHHel1EyyEJs1fWNU3D5JIGlgzcPAV
+ZlbQyUCZXf0oslSa+HE85fO/
FRq9SVSBSHrmbeb0frlPhgMzgSmqLeyhlbr6wwWIDbREXAMPLEJZ49H7RdQxdKyYrZPWvRgcr0qI2EL0tAajnpQQ8UZq
Aqter0xN5PhFL0J49OWTacwCGRAjLhibAx7K1t/1ZXWo6c+ijq8clll327EXAMPLE/
e89GC89KcmKCxfGQniDAUgF8UqofIbq3ZOUgiAAYCVXclI4L68NhVXyoWuQXPBRQSEXAMPLEWm74tDL9tFN3c7tSe/
Oz0cTR+4sAAAIPAAAAB3NzaC1yc2EAAAIAQnG/
L0DqiSnLrWhEox4aHqMgd0m0oLLAYx6OQH9F0TM9EXAMPLE961rzSCMon7ZgsWNnL0OwZQgDG
+rtJ4N0B7HOVwns4ynUFbzNQ3qFGGeE3lKwX1L41vV1iSy7sDk8aI0LmrKJi1LE1Qc1l8uboRlwoXOYEXAMPLEaUCeX
+10+WEXAMPLEg6Y4U4ZvE2B3xyRdpvysb5TGFNtk5qPslacnVkoLOGsZZXMpLGJnG4OBpQLLtpj9sNMxAgZPCAUjhkqk
+nxO9O4NUZ2pTWbVSUaV1gm6pug9xbwNO1Im21t34JeLlKTqxcJ6zzS8W0c0KKpAm5c4hWkseMbyutS2jav/4hiS
+BhrYgptzfwe5qRXEXAMPLEHZQr3YfGzYoBJ/
lLK3NHhxOihhsfAYwMei0BFZT1F/7CT3IH4iitEkIgodi06/
Mw6UDqMPozyQCK1lEA6LFhYCOZG9drWcoRa74lM4kY9TP028Za8gDMh1WpkXLq9Gixon5OHP8aM/
sEXAMPLEr2+fnkw+1BtoO5L6+VKoPlXaGqZ/fBYEXAMPLEAMQHjnLM1JYNvtEEPhp+TNzXHzuixWf/
Ht04m0AVpXrzIDXaS1O2tXY=",
        "ipAddress": "192.0.2.0",
        "privateKey": "-----BEGIN RSA PRIVATE KEY-----
\nEXAMPLEBAAKCAQEA+AD3qeU2toBy5O5v7wnRLVo/tngVickL5+6Jf4tPrPeuoebM
\nfKlA+/ZTwe6uVBEneVWRhbcra8pH0CZ44sKnuxFeWoM7425S49uhW9+xCnWvR1Xw
```

```
\njrvKVm75Mu08p/cNvfWugrBuaPB65DspgxNnOfZWMVxpIpSqOSPWmSwQHV597d6C
\nrEXAMPLEo8hJmqz2KFQ09X7fB2lBruGgr9aXiNPmWmovYKqwFmrnFvR7odFmDecq
\n5EXAMPLE9dyU1ZsrWhGby77eYrVaFl0GNGQ8qy1HGUiscquZ9NDIL49n4mXbfsTH
\n0EXAMPLE12ZqsfLiYnSaUYCwjE74qH8ECVPytQIDAQABAoIBAHeZV9Z58JHAjifz
\nCEXAMPLEEqC3doOVDgXSlkKI92qNo4z2VcUEho878paCuVVXVHcCGgSnGeyIh2tN
\nMEXAMPLESohR427BhH3YLA+3Z5SIvnejbTgYPfLC37B8khTaYqkqMvdZiFVZK5qn
\nIEXAMPLEM93oF9eSZCjcLKB/jGHsfb0eCDMP8BshHE2beuqzVMoK1DxOnvoP3+Fp
\nAEXAMPLESq6pDpCo9YVUX8g1u3Ro9cPl2LXHDy+oVEY5KhbZQJ7VU1I72WOvppWW
\nOEXAMPLEkgYlq7p6qYtYcSgTEjz14gDiMfQ7SyHB3alkIoNONQ9ZPaWHyJvymeud
\noQTNuz0CgYEA/LFWNTEZrzdzdR1kJmyNRmAermU0B6utyNENChAlHGSHkB+1lVSh
\nbEXAMPLEQo9ooUeW5UxO3YwacZLoDT1mwxw1Ptc1+PNycZoLe1fE9UdARrdmGTob
\n8l7CPLSXp3xuR8VqSp2fnIc7hfiQs/NrPX9gm/EOrB0we0RKyDSzWScCgYEA+z/r
\niob+nJZq0YbnOSuP6oMULP4vnWniWj8MIhUJU53LwSAM8DeJdONKDdkuiOd52aAL
\nVgn7nLo88rVWKhJwVc4tu/rNgZLcR3bP4+kL6zand0KQnMLyOzNA2Ys26aa5udH1\nqWl0WTt9WEm/
h10ndC1knOMectrvsG17b38y5sMCgYEA54NiRGGz8oCPW6GN/FZA
\nKEXAMPLE5tw34GEH3Uxlc9n3CejDaQmczOATwX4nIwRZDEqWyYZcS0btg1jhGiBD\nYEXAMPLEkc8Z71L/
agZEAaVCEog9FqfSqwB
+XTfoKh8qur74X1yCu9p6gof1q6k9\neEXAMPLEchJcNNOg4ETIfMkCgYBdVORRhE4mqvWpOdzA7v66FdEz2YSkjAXKk
\naEXAMPLE8Z/8yBSmuBv1Qv03XA12my462uB92uzzGAuW
+1yBc2Kn1sXqYTy0y1z0\ngEXAMPLEBogjw4MqHKL1bPKMHyQU8/
q24PaYgzHPzy13wlH6pTYf1XqlHdE2D6Vv\nyEXAMPLEgQC3i/
kVVhky/2XRwRVlC7JO2Bg3QGTx38hpmDa5IuofKANjA+Wa3/zy\nbEXAMPLE6ytQgD9GN/YtBq+uhO
+2ZkvXPL+CWRi0ZRXpPwYDBBFU9Cw0AuWWGlL8\nwEXAMPLExMlcysRgcWB9RNgf3AuOpFd2i6XT/
riNsvvkpmJ+VooU8g==\n-----END RSA PRIVATE KEY-----\n",
        "protocol": "ssh",
        "instanceName": "WordPress_Multisite-1",
        "username": "bitnami",
        "hostKeys": [
            {
                "algorithm": "ssh-rsa",
                "publicKey":
 "AEXAMPLEaC1yc2EAAAADAQABAAABAQCoeR9ieZTjQ3pXCHczuAYZFjlF7t
+uBkXuqeGMRex78pCvmS+DiEXAMPLEuJ1Q8dcKhrQL4HpXbD9dosVCTaJnJwb4MQqsuSVFdHFzy3guP
+BKclWqtxJEXAMPLEsBGqZZlrIv6a9bTA0TCplZ8AD+hSRTaSXXqg6FT
+Qf16IktH0XlMs7xIEXAMPLEmNtjCpzZiGXDHzytoMvUgwa8uHPp44Og36EUu4VqQxoUHPJKoXvcQizyk3K8ym0hP0Tp
Ot6y9HwvykEXAMPLEAfbKjbR42+u6+OSlkr4d339q2U1sTDytJhhs8HUel1wTfGRfp",
                "witnessedAt": 1570744377.699,
                "fingerprintSHA1": "SHA1:GEXAMPLEMoYgUg0ucadqU9Bt3Lk",
                "fingerprintSHA256": "SHA256:IEXAMPLEcB5vgxnAUoJawbdZ
+MwELhIp6FUxuwq/LIU"
            },
            {
                "algorithm": "ssh-ed25519",
```

```
                "publicKey":
 "AEXAMPLEaC1lZDI1NTE5AAAAIC1gwGPDfGaONxEXAMPLEJX3UNap781QxHQmn8nzlrUv",
                "witnessedAt": 1570744377.697,
                "fingerprintSHA1": "SHA1:VEXAMPLE5ReqSmTgv03sSUw9toU",
                "fingerprintSHA256": "SHA256:0EXAMPLEdE6tI95k3TJpG
+qhJbAoknB0yz9nAEaDt3A"
            },
            {
                "algorithm": "ecdsa-sha2-nistp256",
                "publicKey":
 "AEXAMPLEZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABEXAMPLE9B4mZy8YSsZW7cixCDq5yHSAAxjJkDo5
+EnKlDCsYtUkxxEXAMPLE6VOWL2z63RTKa2AUPgd8irjxWI=",
                "witnessedAt": 1570744377.707,
                "fingerprintSHA1": "SHA1:UEXAMPLEOYCfXsCf2G6tDg+7YG0",
                "fingerprintSHA256": "SHA256:wEXAMPLEQ9a/
iEXAMPLEhRufm6U9vFU4cpkMPHnBsNA"
            }
        ]
    }
}
```

- 자세한 API 내용은 명령 참조GetInstanceAccessDetails의 섹션을 참조하세요. AWS CLI

### get-instance-metric-data

다음 코드 예시에서는 get-instance-metric-data을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스의 지표 데이터를 가져오려면

다음 get-instance-metric-data 예제는 인스턴스 와 같이 1571342400 및 사이의 7200 초(2시간)CPUUtilization당 평균 비율을 반환1571428800합니다MEAN-1.

unix 시간 변환기를 사용하여 시작 및 종료 시간을 식별하는 것이 좋습니다.

```
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
```

출력:


```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetInstanceMetricData의 섹션을 참조하세요. AWS CLI

### get-instance-port-states

다음 코드 예시에서는 get-instance-port-states을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스에 대한 방화벽 정보를 가져오려면

다음 get-instance-port-states 예제에서는 인스턴스 에 대해 구성된 방화벽 포트를 반환합니다MEAN-1.

```
aws lightsail get-instance-port-states \
    --instance-name MEAN-1
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetInstancePortStates의 섹션을 참조하세요. AWS CLI

### get-instance-snapshot

다음 코드 예시에서는 get-instance-snapshot을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 인스턴스 스냅샷에 대한 정보를 가져오려면

다음 get-instance-snapshot 예제에서는 지정된 인스턴스 스냅샷에 대한 세부 정보를 표시합니다.

```
aws lightsail get-instance-snapshot \
    --instance-snapshot-name MEAN-1-1571419854
```

출력:

```
{
    "instanceSnapshot": {
        "name": "MEAN-1-1571419854",
        "arn": "arn:aws:lightsail:us-west-2:111122223333:InstanceSnapshot/
ac54700c-48a8-40fd-b065-2EXAMPLEac8f",
        "supportCode": "6EXAMPLE3362/ami-0EXAMPLE67a73020d",
        "createdAt": 1571419891.927,
        "location": {
            "availabilityZone": "all",
            "regionName": "us-west-2"
        },
        "resourceType": "InstanceSnapshot",
        "tags": [],
        "state": "available",
        "fromAttachedDisks": [],
        "fromInstanceName": "MEAN-1",
        "fromInstanceArn": "arn:aws:lightsail:us-west-2:111122223333:Instance/
bd470fc5-a68b-44c5-8dbc-8EXAMPLEbada",
        "fromBlueprintId": "mean",
        "fromBundleId": "medium_3_0",
        "isFromAutoSnapshot": false,
        "sizeInGb": 80
    }
}
```

- 자세한 API 내용은 명령 참조GetInstanceSnapshot의 섹션을 참조하세요. AWS CLI

### `get-instance-snapshots`

다음 코드 예시에서는 `get-instance-snapshots`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 인스턴스 스냅샷에 대한 정보를 가져오려면

다음 `get-instance-snapshots` 예제에서는 구성된 AWS 리전의 모든 인스턴스 스냅샷에 대한 세부 정보를 표시합니다.

```
aws lightsail get-instance-snapshots
```

출력:

```
{
    "instanceSnapshots": [
        {
            "name": "MEAN-1-1571421498",
            "arn": "arn:aws:lightsail:us-west-2:111122223333:InstanceSnapshot/
a20e6ebe-b0ee-4ae4-a750-3EXAMPLEcb0c",
            "supportCode": "6EXAMPLE3362/ami-0EXAMPLEe33cabfa1",
            "createdAt": 1571421527.755,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "resourceType": "InstanceSnapshot",
            "tags": [
                {
                    "key": "no_delete"
                }
            ],
            "state": "available",
            "fromAttachedDisks": [],
            "fromInstanceName": "MEAN-1",
            "fromInstanceArn": "arn:aws:lightsail:us-
west-2:111122223333:Instance/1761aa0a-6038-4f25-8b94-2EXAMPLE19fd",
            "fromBlueprintId": "wordpress",
            "fromBundleId": "micro_3_0",
            "isFromAutoSnapshot": false,
            "sizeInGb": 40
        },
        {
            "name": "MEAN-1-1571419854",
            "arn": "arn:aws:lightsail:us-west-2:111122223333:InstanceSnapshot/
ac54700c-48a8-40fd-b065-2EXAMPLEac8f",
            "supportCode": "6EXAMPLE3362/ami-0EXAMPLE67a73020d",
            "createdAt": 1571419891.927,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "resourceType": "InstanceSnapshot",
            "tags": [],
            "state": "available",
            "fromAttachedDisks": [],
            "fromInstanceName": "MEAN-1",
```

```
            "fromInstanceArn": "arn:aws:lightsail:us-west-2:111122223333:Instance/
bd470fc5-a68b-44c5-8dbc-8EXAMPLEbada",
            "fromBlueprintId": "mean",
            "fromBundleId": "medium_3_0",
            "isFromAutoSnapshot": false,
            "sizeInGb": 80
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetInstanceSnapshots의 섹션을 참조하세요. AWS CLI

### get-instance-state

다음 코드 예시에서는 get-instance-state을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스 상태에 대한 정보를 가져오려면

다음 get-instance-state 예제에서는 지정된 인스턴스의 상태를 반환합니다.

```
aws lightsail get-instance-state \
    --instance-name MEAN-1
```

출력:

```
{
    "state": {
        "code": 16,
        "name": "running"
    }
}
```

- 자세한 API 내용은 명령 참조GetInstanceState의 섹션을 참조하세요. AWS CLI

### get-instance

다음 코드 예시에서는 get-instance을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스에 대한 정보를 가져오려면

다음 `get-instance` 예제에서는 인스턴스 에 대한 세부 정보를 표시합니다`MEAN-1`.

```
aws lightsail get-instance \
    --instance-name MEAN-1
```

출력:

```
{
    "instance": {
        "name": "MEAN-1",
        "arn": "arn:aws:lightsail:us-west-2:111122223333:Instance/bd470fc5-
a68b-44c5-8dbc-EXAMPLE4bada",
        "supportCode": "6EXAMPLE3362/i-05EXAMPLE407c97d3",
        "createdAt": 1570635023.124,
        "location": {
            "availabilityZone": "us-west-2a",
            "regionName": "us-west-2"
        },
        "resourceType": "Instance",
        "tags": [],
        "blueprintId": "mean",
        "blueprintName": "MEAN",
        "bundleId": "medium_3_0",
        "isStaticIp": false,
        "privateIpAddress": "192.0.2.0",
        "publicIpAddress": "192.0.2.0",
        "hardware": {
            "cpuCount": 2,
            "disks": [
                {
                    "createdAt": 1570635023.124,
                    "sizeInGb": 80,
                    "isSystemDisk": true,
                    "iops": 240,
                    "path": "/dev/xvda",
                    "attachedTo": "MEAN-1",
                    "attachmentState": "attached"
                }
            ],
```

```
            "ramSizeInGb": 4.0
        },
        "networking": {
            "monthlyTransfer": {
                "gbPerMonthAllocated": 4096
            },
            "ports": [
                {
                    "fromPort": 80,
                    "toPort": 80,
                    "protocol": "tcp",
                    "accessFrom": "Anywhere (0.0.0.0/0)",
                    "accessType": "public",
                    "commonName": "",
                    "accessDirection": "inbound"
                },
                {
                    "fromPort": 22,
                    "toPort": 22,
                    "protocol": "tcp",
                    "accessFrom": "Anywhere (0.0.0.0/0)",
                    "accessType": "public",
                    "commonName": "",
                    "accessDirection": "inbound"
                },
                {
                    "fromPort": 443,
                    "toPort": 443,
                    "protocol": "tcp",
                    "accessFrom": "Anywhere (0.0.0.0/0)",
                    "accessType": "public",
                    "commonName": "",
                    "accessDirection": "inbound"
                }
            ]
        },
        "state": {
            "code": 16,
            "name": "running"
        },
        "username": "bitnami",
        "sshKeyName": "MyKey"
    }
```

```
}
```

- 자세한 API 내용은 명령 참조GetInstance의 섹션을 참조하세요. AWS CLI

### `get-instances`

다음 코드 예시에서는 `get-instances`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 인스턴스에 대한 정보를 가져오려면

다음 `get-instances` 예제에서는 구성된 AWS 리전의 모든 인스턴스에 대한 세부 정보를 표시합니다.

```
aws lightsail get-instances
```

출력:

```
{
    "instances": [
        {
            "name": "Windows_Server_2022-1",
            "arn": "arn:aws:lightsail:us-
west-2:111122223333:Instance/0f44fbb9-8f55-4e47-a25e-EXAMPLE04763",
            "supportCode": "62EXAMPLE362/i-0bEXAMPLE71a686b9",
            "createdAt": 1571332358.665,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "resourceType": "Instance",
            "tags": [],
            "blueprintId": "windows_server_2022",
            "blueprintName": "Windows Server 2022",
            "bundleId": "large_win_3_0",
            "isStaticIp": false,
            "privateIpAddress": "192.0.2.0",
            "publicIpAddress": "192.0.2.0",
            "hardware": {
                "cpuCount": 1,
                "disks": [
```

```
                    {
                        "createdAt": 1571332358.665,
                        "sizeInGb": 160,
                        "isSystemDisk": true,
                        "iops": 180,
                        "path": "/dev/sda1",
                        "attachedTo": "Windows_Server_2022-1",
                        "attachmentState": "attached"
                    },
                    {
                        "name": "my-disk-for-windows-server",
                        "arn": "arn:aws:lightsail:us-
west-2:111122223333:Disk/4123a81c-484c-49ea-afea-5EXAMPLEda87",
                        "supportCode": "6EXAMPLE3362/vol-0EXAMPLEb2b99ca3d",
                        "createdAt": 1571355063.494,
                        "location": {
                            "availabilityZone": "us-west-2a",
                            "regionName": "us-west-2"
                        },
                        "resourceType": "Disk",
                        "tags": [],
                        "sizeInGb": 128,
                        "isSystemDisk": false,
                        "iops": 384,
                        "path": "/dev/xvdf",
                        "state": "in-use",
                        "attachedTo": "Windows_Server_2022-1",
                        "isAttached": true,
                        "attachmentState": "attached"
                    }
                ],
                "ramSizeInGb": 8.0
            },
            "networking": {
                "monthlyTransfer": {
                    "gbPerMonthAllocated": 3072
                },
                "ports": [
                    {
                        "fromPort": 80,
                        "toPort": 80,
                        "protocol": "tcp",
                        "accessFrom": "Anywhere (0.0.0.0/0)",
                        "accessType": "public",
```

```
                        "commonName": "",
                        "accessDirection": "inbound"
                    },
                    {
                        "fromPort": 22,
                        "toPort": 22,
                        "protocol": "tcp",
                        "accessFrom": "Anywhere (0.0.0.0/0)",
                        "accessType": "public",
                        "commonName": "",
                        "accessDirection": "inbound"
                    },
                    {
                        "fromPort": 3389,
                        "toPort": 3389,
                        "protocol": "tcp",
                        "accessFrom": "Anywhere (0.0.0.0/0)",
                        "accessType": "public",
                        "commonName": "",
                        "accessDirection": "inbound"
                    }
                ]
            },
            "state": {
                "code": 16,
                "name": "running"
            },
            "username": "Administrator",
            "sshKeyName": "LightsailDefaultKeyPair"
        },
        {
            "name": "MEAN-1",
            "arn": "arn:aws:lightsail:us-west-2:111122223333:Instance/bd470fc5-
a68b-44c5-8dbc-8EXAMPLEbada",
            "supportCode": "6EXAMPLE3362/i-0EXAMPLEa407c97d3",
            "createdAt": 1570635023.124,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "resourceType": "Instance",
            "tags": [],
            "blueprintId": "mean",
            "blueprintName": "MEAN",
```

```
            "bundleId": "medium_3_0",
            "isStaticIp": false,
            "privateIpAddress": "192.0.2.0",
            "publicIpAddress": "192.0.2.0",
            "hardware": {
                "cpuCount": 2,
                "disks": [
                    {
                        "name": "Disk-1",
                        "arn": "arn:aws:lightsail:us-west-2:111122223333:Disk/
c21cfb0a-07f2-44ae-9a23-bEXAMPLE8096",
                        "supportCode": "6EXAMPLE3362/vol-0EXAMPLEf2f88b32f",
                        "createdAt": 1566585439.587,
                        "location": {
                            "availabilityZone": "us-west-2a",
                            "regionName": "us-west-2"
                        },
                        "resourceType": "Disk",
                        "tags": [
                            {
                                "key": "test"
                            }
                        ],
                        "sizeInGb": 8,
                        "isSystemDisk": false,
                        "iops": 240,
                        "path": "/dev/xvdf",
                        "state": "in-use",
                        "attachedTo": "MEAN-1",
                        "isAttached": true,
                        "attachmentState": "attached"
                    },
                    {
                        "createdAt": 1570635023.124,
                        "sizeInGb": 80,
                        "isSystemDisk": true,
                        "iops": 240,
                        "path": "/dev/sda1",
                        "attachedTo": "MEAN-1",
                        "attachmentState": "attached"
                    }
                ],
                "ramSizeInGb": 4.0
            },
```

```
            "networking": {
                "monthlyTransfer": {
                    "gbPerMonthAllocated": 4096
                },
                "ports": [
                    {
                        "fromPort": 80,
                        "toPort": 80,
                        "protocol": "tcp",
                        "accessFrom": "Anywhere (0.0.0.0/0)",
                        "accessType": "public",
                        "commonName": "",
                        "accessDirection": "inbound"
                    },
                    {
                        "fromPort": 22,
                        "toPort": 22,
                        "protocol": "tcp",
                        "accessFrom": "Anywhere (0.0.0.0/0)",
                        "accessType": "public",
                        "commonName": "",
                        "accessDirection": "inbound"
                    },
                    {
                        "fromPort": 443,
                        "toPort": 443,
                        "protocol": "tcp",
                        "accessFrom": "Anywhere (0.0.0.0/0)",
                        "accessType": "public",
                        "commonName": "",
                        "accessDirection": "inbound"
                    }
                ]
            },
            "state": {
                "code": 16,
                "name": "running"
            },
            "username": "bitnami",
            "sshKeyName": "MyTestKey"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetInstances의 섹션을 참조하세요. AWS CLI

### get-key-pair

다음 코드 예시에서는 get-key-pair을 사용하는 방법을 보여 줍니다.

AWS CLI

키 페어에 대한 정보를 가져오려면

다음 get-key-pair 예제에서는 지정된 키 페어에 대한 세부 정보를 표시합니다.

```
aws lightsail get-key-pair \
    --key-pair-name MyKey1
```

출력:

```
{
    "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"
    }
}
```

- 자세한 API 내용은 명령 참조GetKeyPair의 섹션을 참조하세요. AWS CLI

### get-key-pairs

다음 코드 예시에서는 get-key-pairs을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 키 페어에 대한 정보를 가져오려면

다음 `get-key-pairs` 예제에서는 구성된 AWS 리전의 모든 키 페어에 대한 세부 정보를 표시합니다.

```
aws lightsail get-key-pairs
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetKeyPairs의 섹션을 참조하세요. AWS CLI

### `get-load-balancer-tls-certificates`

다음 코드 예시에서는 `get-load-balancer-tls-certificates`을 사용하는 방법을 보여 줍니다.

AWS CLI

로드 밸런서의 TLS 인증서에 대한 정보를 가져오려면

다음 get-load-balancer-tls-certificates 예제에서는 지정된 로드 밸런서의 TLS 인증서에 대한 세부 정보를 표시합니다.

```
aws lightsail get-load-balancer-tls-certificates \
    --load-balancer-name LoadBalancer-1
```

출력:

```
{
    "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"
            ]
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetLoadBalancerTlsCertificates의 섹션을 참조하세요. AWS CLI

### get-load-balancer

다음 코드 예시에서는 get-load-balancer을 사용하는 방법을 보여 줍니다.

AWS CLI

로드 밸런서에 대한 정보를 가져오려면

다음 get-load-balancer 예제에서는 지정된 로드 밸런서에 대한 세부 정보를 표시합니다.

```
aws lightsail get-load-balancer \
    --load-balancer-name LoadBalancer-1
```

출력:

```
{
    "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"
        }
    }
}
```

- 자세한 API 내용은 명령 참조GetLoadBalancer의 섹션을 참조하세요. AWS CLI

### `get-load-balancers`

다음 코드 예시에서는 `get-load-balancers`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 로드 밸런서에 대한 정보를 가져오려면

다음 get-load-balancers 예제에서는 구성된 AWS 리전의 모든 로드 밸런서에 대한 세부 정보를 표시합니다.

```
aws lightsail get-load-balancers
```

출력:

```
{
    "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"
            }
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetLoadBalancers의 섹션을 참조하세요. AWS CLI

### get-operation

다음 코드 예시에서는 get-operation을 사용하는 방법을 보여 줍니다.

AWS CLI

단일 작업에 대한 정보를 가져오려면

다음 get-operation 예제에서는 지정된 작업에 대한 세부 정보를 표시합니다.

```
aws lightsail get-operation \
    --operation-id e5700e8a-daf2-4b49-bc01-3EXAMPLE910a
```

출력:

```
{
    "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
    }
}
```

- 자세한 API 내용은 명령 참조GetOperation의 섹션을 참조하세요. AWS CLI

### get-operations-for-resource

다음 코드 예시에서는 get-operations-for-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 대한 모든 작업을 가져오려면

다음 get-operations-for-resource 예제에서는 지정된 리소스의 모든 작업에 대한 세부 정보를 표시합니다.

```
aws lightsail get-operations-for-resource \
    --resource-name LoadBalancer-1
```

출력:

```
{
    "operations": [
        {
            "id": "e2973046-43f8-4252-a4b4-9EXAMPLE69ce",
            "resourceName": "LoadBalancer-1",
            "resourceType": "LoadBalancer",
            "createdAt": 1571678786.071,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationDetails": "MEAN-1",
            "operationType": "DetachInstancesFromLoadBalancer",
```

```
            "status": "Succeeded",
            "statusChangedAt": 1571679087.57
        },
        {
            "id": "2d742a18-0e7f-48c8-9705-3EXAMPLEf98a",
            "resourceName": "LoadBalancer-1",
            "resourceType": "LoadBalancer",
            "createdAt": 1571678782.784,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationDetails": "MEAN-1",
            "operationType": "AttachInstancesToLoadBalancer",
            "status": "Succeeded",
            "statusChangedAt": 1571678798.465
        },
        {
            "id": "6c700fcc-4246-40ab-952b-1EXAMPLEdac2",
            "resourceName": "LoadBalancer-1",
            "resourceType": "LoadBalancer",
            "createdAt": 1571678775.297,
            "location": {
                "availabilityZone": "all",
                "regionName": "us-west-2"
            },
            "isTerminal": true,
            "operationDetails": "MEAN-3",
            "operationType": "AttachInstancesToLoadBalancer",
            "status": "Succeeded",
            "statusChangedAt": 1571678842.806
        },
        ...
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetOperationsForResource의 섹션을 참조하세요. AWS CLI

### `get-operations`

다음 코드 예시에서는 `get-operations`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 작업에 대한 정보를 가져오려면

다음 get-operations 예제에서는 구성된 AWS 리전의 모든 작업에 대한 세부 정보를 표시합니다.

```
aws lightsail get-operations
```

출력:

```
{
    "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
        },
        ...
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetOperations의 섹션을 참조하세요. AWS CLI

### get-regions

다음 코드 예시에서는 get-regions을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon Lightsail의 모든 AWS 리전을 가져오려면

다음 get-regions 예제에서는 Amazon Lightsail 의 모든 AWS 리전에 대한 세부 정보를 표시합니다.

```
aws lightsail get-regions
```

출력:

```
{
    "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": []
        },
        ...
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetRegions의 섹션을 참조하세요. AWS CLI

### `get-relational-database-blueprints`

다음 코드 예시에서는 `get-relational-database-blueprints`을 사용하는 방법을 보여 줍니다.

AWS CLI

새 관계형 데이터베이스에 대한 청사진을 가져오려면

다음 `get-relational-database-blueprints` 예제에서는 Amazon Lightsail 에서 새 관계형 데이터베이스를 생성하는 데 사용할 수 있는 모든 관계형 데이터베이스 청사진에 대한 세부 정보를 보여줍니다.

```
aws lightsail get-relational-database-blueprints
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetRelationalDatabaseBlueprints의 섹션을 참조하세요. AWS CLI

### `get-relational-database-bundles`

다음 코드 예시에서는 `get-relational-database-bundles`을 사용하는 방법을 보여 줍니다.

AWS CLI

새 관계형 데이터베이스의 번들을 가져오려면

다음 `get-relational-database-bundles` 예제에서는 Amazon Lightsail 에서 새 관계형 데이터베이스를 생성하는 데 사용할 수 있는 모든 관계형 데이터베이스 번들에 대한 세부 정보를 보여 줍니다. `--include-inactive` 플래그가 명령에 지정되지 않았으므로 응답에 비활성 번들이 포함되지 않습니다. 비활성 번들을 사용하여 새 관계형 데이터베이스를 생성할 수 없습니다.

```
aws lightsail get-relational-database-bundles
```

출력:

```
{
    "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
        }
    ]
}
```


자세한 내용은 Amazon Lightsail 개발자 안내서의 Amazon Lightsail에서 데이터베이스 생성을 참조하세요. Amazon Lightsail

- 자세한 API 내용은 명령 참조GetRelationalDatabaseBundles의 섹션을 참조하세요. AWS CLI

### get-relational-database-events

다음 코드 예시에서는 get-relational-database-events을 사용하는 방법을 보여 줍니다.

AWS CLI

관계형 데이터베이스의 이벤트를 가져오려면

다음 get-relational-database-events 예제에서는 지정된 관계형 데이터베이스에 대해 지난 17시간(10~20분) 동안의 이벤트에 대한 세부 정보를 표시합니다.

```
aws lightsail get-relational-database-events \
    --relational-database-name Database-1 \
    --duration-in-minutes 1020
```

출력:

```
{
    "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"
            ]
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetRelationalDatabaseEvents의 섹션을 참조하세요. AWS CLI

### get-relational-database-log-events

다음 코드 예시에서는 get-relational-database-log-events을 사용하는 방법을 보여 줍니다.

AWS CLI

관계형 데이터베이스의 로그 이벤트를 가져오려면

다음 get-relational-database-log-events 예제에서는 관계형 데이터베이스 에 대해 1570733176 와 간의 지정된 로그에 1571597176 대한 세부 정보를 표시합니다Database1. 반환된 정보는 에서 시작하도록 구성됩니다head.

unix 시간 변환기를 사용하여 시작 및 종료 시간을 식별하는 것이 좋습니다.

```
aws lightsail get-relational-database-log-events \
    --relational-database-name Database1 \
    --log-stream-name error \
    --start-from-head \
    --start-time 1570733176 \
    --end-time 1571597176
```

출력:

```
{
    "resourceLogEvents": [
        {
            "createdAt": 1570820267.0,
            "message": "2019-10-11 18:57:47 20969 [Warning] IP address '192.0.2.0'
 could not be resolved: Name or service not known"
        },
        {
            "createdAt": 1570860974.0,
            "message": "2019-10-12 06:16:14 20969 [Warning] IP address '8192.0.2.0'
 could not be resolved: Temporary failure in name resolution"
        },
        {
            "createdAt": 1570860977.0,
            "message": "2019-10-12 06:16:17 20969 [Warning] IP address '192.0.2.0'
 could not be resolved: Temporary failure in name resolution"
        },
        {
            "createdAt": 1570860979.0,
```

```
            "message": "2019-10-12 06:16:19 20969 [Warning] IP address '192.0.2.0'
 could not be resolved: Temporary failure in name resolution"
        },
        {
            "createdAt": 1570860981.0,
            "message": "2019-10-12 06:16:21 20969 [Warning] IP address '192.0.2.0'
 could not be resolved: Temporary failure in name resolution"
        },
        {
            "createdAt": 1570860982.0,
            "message": "2019-10-12 06:16:22 20969 [Warning] IP address '192.0.2.0'
 could not be resolved: Temporary failure in name resolution"
        },
        {
            "createdAt": 1570860984.0,
            "message": "2019-10-12 06:16:24 20969 [Warning] IP address '192.0.2.0'
 could not be resolved: Temporary failure in name resolution"
        },
        {
            "createdAt": 1570860986.0,
            "message": "2019-10-12 06:16:26 20969 [Warning] IP address '192.0.2.0'
 could not be resolved: Temporary failure in name resolution"
        },
        ...
        }
    ],
    "nextBackwardToken":
 "eEXAMPLEZXJUZXh0IjoiZnRWb3F3cUpRSlQ5NndMYThxelRUZlFhR3J6c2dKWEEvM2kvajZMZzVVVWpqRDN0YjFXTj
    "nextForwardToken":
 "eEXAMPLEZXJUZXh0IjoiT09Lb0Z6ZFRJbHhaNEQ5N2tPbkkwRmwwNUxPZjFTbFFwUklQbzlSaWgvMWVXbEk4aG56VH
}
```

- 자세한 API 내용은 명령 참조GetRelationalDatabaseLogEvents의 섹션을 참조하세요. AWS CLI

### `get-relational-database-log-streams`

다음 코드 예시에서는 `get-relational-database-log-streams`을 사용하는 방법을 보여 줍니다.

AWS CLI

관계형 데이터베이스의 로그 스트림을 가져오려면

다음 get-relational-database-log-streams 예제에서는 지정된 관계형 데이터베이스에 사용 가능한 모든 로그 스트림을 반환합니다.

```
aws lightsail get-relational-database-log-streams \
--relational-database-name Database1
```

출력:

```
{
    "logStreams": [
        "audit",
        "error",
        "general",
        "slowquery"
    ]
}
```

- 자세한 API 내용은 명령 참조GetRelationalDatabaseLogStreams의 섹션을 참조하세요. AWS CLI

### get-relational-database-master-user-password

다음 코드 예시에서는 get-relational-database-master-user-password을 사용하는 방법을 보여 줍니다.

AWS CLI

관계형 데이터베이스의 마스터 사용자 암호를 가져오려면

다음 get-relational-database-master-user-password 예제에서는 지정된 관계형 데이터베이스의 마스터 사용자 암호에 대한 정보를 반환합니다.

```
aws lightsail get-relational-database-master-user-password \
    --relational-database-name Database-1
```

출력:

```
{
    "masterUserPassword": "VEXAMPLEec.9qvx,_t<)Wkf)kwboM,>2",
    "createdAt": 1571259453.959
}
```

- 자세한 API 내용은 명령 참조GetRelationalDatabaseMasterUserPassword의 섹션을 참조하세요. AWS CLI

### get-relational-database-metric-data

다음 코드 예시에서는 get-relational-database-metric-data을 사용하는 방법을 보여 줍니다.

AWS CLI

관계형 데이터베이스의 지표 데이터를 가져오려면

다음 get-relational-database-metric-data 예제에서는 관계형 데이터베이스 에 1571597176 대해 1570733176 와 사이의 24시간(86400초) 기간 DatabaseConnections 동안 지표의 수 합계를 반환합니다Database1.

unix 시간 변환기를 사용하여 시작 및 종료 시간을 식별하는 것이 좋습니다.

```
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
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetRelationalDatabaseMetricData의 섹션을 참조하세요. AWS CLI

### get-relational-database-parameters

다음 코드 예시에서는 get-relational-database-parameters을 사용하는 방법을 보여 줍니다.

AWS CLI

관계형 데이터베이스의 파라미터를 가져오려면

다음 get-relational-database-parameters 예제에서는 지정된 관계형 데이터베이스에 사용 가능한 모든 파라미터에 대한 정보를 반환합니다.

```
aws lightsail get-relational-database-parameters \
    --relational-database-name Database-1
```

출력:

```
{
    "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"
        },
        ...
        }
    ]
}
```

자세한 내용은 Lightsail Dev Guide의 Amazon Lightsail에서 데이터베이스 파라미터 업데이트를 참조하세요.

- 자세한 API 내용은 명령 참조GetRelationalDatabaseParameters의 섹션을 참조하세요. AWS CLI

### `get-relational-database-snapshot`

다음 코드 예시에서는 `get-relational-database-snapshot`을 사용하는 방법을 보여 줍니다.

AWS CLI

관계형 데이터베이스 스냅샷에 대한 정보를 가져오려면

다음 `get-relational-database-snapshot` 예제에서는 지정된 관계형 데이터베이스 스냅샷에 대한 세부 정보를 표시합니다.

```
aws lightsail get-relational-database-snapshot \
    --relational-database-snapshot-name Database-1-1571350042
```

출력:

```
{
    "relationalDatabaseSnapshot": {
```

```
        "name": "Database-1-1571350042",
        "arn": "arn:aws:lightsail:us-
west-2:111122223333:RelationalDatabaseSnapshot/0389bbad-4b85-4c3d-9EXAMPLEaee3643d2",
        "supportCode": "6EXAMPLE3362/ls-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"
    }
}
```

- 자세한 API 내용은 명령 참조GetRelationalDatabaseSnapshot의 섹션을 참조하세요. AWS CLI

### get-relational-database-snapshots

다음 코드 예시에서는 get-relational-database-snapshots을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 관계형 데이터베이스 스냅샷에 대한 정보를 가져오려면

다음 get-relational-database-snapshots 예제에서는 구성된 AWS 리전의 모든 관계형 데이터베이스 스냅샷에 대한 세부 정보를 표시합니다.

```
aws lightsail get-relational-database-snapshots
```

출력:

```
{
    "relationalDatabaseSnapshots": [
```

```
        {
            "name": "Database-1-1571350042",
            "arn": "arn:aws:lightsail:us-
west-2:111122223333:RelationalDatabaseSnapshot/0389bbad-4b85-4c3d-9861-6EXAMPLE43d2",
            "supportCode": "6EXAMPLE3362/
ls-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/
ls-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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetRelationalDatabaseSnapshots의 섹션을 참조하세요. AWS CLI

### get-relational-database

다음 코드 예시에서는 get-relational-database을 사용하는 방법을 보여 줍니다.

AWS CLI

관계형 데이터베이스에 대한 정보를 가져오려면

다음 get-relational-database 예제에서는 지정된 관계형 데이터베이스에 대한 세부 정보를 표시합니다.

```
aws lightsail get-relational-database \
    --relational-database-name Database-1
```

출력:

```
{
    "relationalDatabase": {
        "name": "Database-1",
        "arn": "arn:aws:lightsail:us-
west-2:111122223333:RelationalDatabase/7ea932b1-b85a-4bd5-9b3e-bEXAMPLE8cc4",
        "supportCode": "6EXAMPLE3362/ls-9EXAMPLE8ad863723b62cc8901a8aa6e794ae0d2",
        "createdAt": 1571259453.795,
        "location": {
            "availabilityZone": "us-west-2a",
            "regionName": "us-west-2"
        },
        "resourceType": "RelationalDatabase",
        "tags": [],
        "relationalDatabaseBlueprintId": "mysql_8_0",
```

```
        "relationalDatabaseBundleId": "micro_1_0",
        "masterDatabaseName": "dbmaster",
        "hardware": {
            "cpuCount": 1,
            "diskSizeInGb": 40,
            "ramSizeInGb": 1.0
        },
        "state": "available",
        "backupRetentionEnabled": false,
        "pendingModifiedValues": {},
        "engine": "mysql",
        "engineVersion": "8.0.16",
        "masterUsername": "dbmasteruser",
        "parameterApplyStatus": "in-sync",
        "preferredBackupWindow": "10:01-10:31",
        "preferredMaintenanceWindow": "sat:11:14-sat:11:44",
        "publiclyAccessible": true,
        "masterEndpoint": {
            "port": 3306,
            "address": "ls-9EXAMPLE8ad863723b62ccEXAMPLEa6e794ae0d2.czowadgeezqi.us-
west-2.rds.amazonaws.com"
        },
        "pendingMaintenanceActions": []
    }
}
```

- 자세한 API 내용은 명령 참조GetRelationalDatabase의 섹션을 참조하세요. AWS CLI

### get-relational-databases

다음 코드 예시에서는 get-relational-databases을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 관계형 데이터베이스에 대한 정보를 가져오려면

다음 get-relational-databases 예제에서는 구성된 AWS 리전의 모든 관계형 데이터베이스에 대한 세부 정보를 표시합니다.

```
aws lightsail get-relational-databases
```

출력:

```
{
    "relationalDatabases": [
        {
            "name": "MySQL",
            "arn": "arn:aws:lightsail:us-
west-2:111122223333:RelationalDatabase/8529020c-3ab9-4d51-92af-5EXAMPLE8979",
            "supportCode": "6EXAMPLE3362/
ls-3EXAMPLEa995d8c3b06b4501356e5f2f28e1aeba",
            "createdAt": 1554306019.155,
            "location": {
                "availabilityZone": "us-west-2a",
                "regionName": "us-west-2"
            },
            "resourceType": "RelationalDatabase",
            "tags": [],
            "relationalDatabaseBlueprintId": "mysql_8_0",
            "relationalDatabaseBundleId": "micro_1_0",
            "masterDatabaseName": "dbmaster",
            "hardware": {
                "cpuCount": 1,
                "diskSizeInGb": 40,
                "ramSizeInGb": 1.0
            },
            "state": "available",
            "backupRetentionEnabled": true,
            "pendingModifiedValues": {},
            "engine": "mysql",
            "engineVersion": "8.0.15",
            "latestRestorableTime": 1571686200.0,
            "masterUsername": "dbmasteruser",
            "parameterApplyStatus": "in-sync",
            "preferredBackupWindow": "07:51-08:21",
            "preferredMaintenanceWindow": "tue:12:18-tue:12:48",
            "publiclyAccessible": true,
            "masterEndpoint": {
                "port": 3306,
                "address":
 "ls-3EXAMPLEa995d8c3b06b4501356e5f2fEXAMPLEa.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-3EXAMPLEb4fffb5cec056220c734713e14bd5fcd",
            "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-3EXAMPLEb4fffb5cec056220c734713eEXAMPLEd.czowadgeezqi.us-
west-2.rds.amazonaws.com"
            },
            "pendingMaintenanceActions": []
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetRelationalDatabases의 섹션을 참조하세요. AWS CLI

### get-static-ip

다음 코드 예시에서는 get-static-ip을 사용하는 방법을 보여 줍니다.

AWS CLI

정적 IP에 대한 정보를 가져오려면

다음 get-static-ip 예제에서는 지정된 정적 IP에 대한 세부 정보를 표시합니다.

```
aws lightsail get-static-ip \
    --static-ip-name StaticIp-1
```

출력:

```
{
    "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
    }
}
```

- 자세한 API 내용은 명령 참조GetStaticIp의 섹션을 참조하세요. AWS CLI

### get-static-ips

다음 코드 예시에서는 get-static-ips을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 정적 IPs

다음 get-static-ips 예제에서는 구성된 AWS 리전의 모든 정적IPs에 대한 세부 정보를 표시합니다.

```
aws lightsail get-static-ips
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetStaticIps의 섹션을 참조하세요. AWS CLI

### is-vpc-peered

다음 코드 예시에서는 is-vpc-peered을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon Lightsail 가상 프라이빗 클라우드가 피어링되었는지 확인하려면

다음 is-vpc-peered 예제에서는 지정된 AWS 리전에 대한 Amazon Lightsail 가상 프라이빗 클라우드(VPC)의 피어링 상태를 반환합니다.

```
aws lightsail is-vpc-peered \
    --region us-west-2
```

출력:

```
{
    "isPeered": true
}
```

- 자세한 API 내용은 명령 참조IsVpcPeered의 섹션을 참조하세요. AWS CLI

### open-instance-public-ports

다음 코드 예시에서는 open-instance-public-ports을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스의 방화벽 포트를 열려면

다음 open-instance-public-ports 예제에서는 지정된 인스턴스에서 TCP 포트 22를 엽니다.

```
aws lightsail open-instance-public-ports \
    --instance-name MEAN-2 \
    --port-info fromPort=22,protocol=TCP,toPort=22
```

출력:

```
{
    "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
    }
}
```

- 자세한 API 내용은 명령 참조OpenInstancePublicPorts의 섹션을 참조하세요. AWS CLI

### `peer-vpc`

다음 코드 예시에서는 `peer-vpc`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon Lightsail 가상 프라이빗 클라우드 피어링

다음 `peer-vpc` 예제는 지정된 AWS 리전에 대해 Amazon Lightsail 가상 프라이빗 클라우드(VPC)를 피어링합니다.

```
aws lightsail peer-vpc \
    --region us-west-2
```

출력:

```
{
    "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
    }
}
```

- 자세한 API 내용은 명령 참조PeerVpc의 섹션을 참조하세요. AWS CLI

### reboot-instance

다음 코드 예시에서는 reboot-instance을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스를 재부팅하려면

다음 reboot-instance 예제에서는 지정된 인스턴스를 재부팅합니다.

```
aws lightsail reboot-instance \
    --instance-name MEAN-1
```

출력:

```
{
    "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
```

```
        }
    ]
}
```

- 자세한 API 내용은 명령 참조RebootInstance의 섹션을 참조하세요. AWS CLI

### `reboot-relational-database`

다음 코드 예시에서는 `reboot-relational-database`을 사용하는 방법을 보여 줍니다.

AWS CLI

관계형 데이터베이스를 재부팅하려면

다음 `reboot-relational-database` 예제에서는 지정된 관계형 데이터베이스를 재부팅합니다.

```
aws lightsail reboot-relational-database \
    --relational-database-name Database-1
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조RebootRelationalDatabase의 섹션을 참조하세요. AWS CLI

### `release-static-ip`

다음 코드 예시에서는 `release-static-ip`을 사용하는 방법을 보여 줍니다.

AWS CLI

정적 IP를 삭제하려면

다음 `release-static-ip` 예제에서는 지정된 정적 IP를 삭제합니다.

```
aws lightsail release-static-ip \
    --static-ip-name StaticIp-1
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ReleaseStaticIp의 섹션을 참조하세요. AWS CLI

### `start-instance`

다음 코드 예시에서는 `start-instance`을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스를 시작하려면

다음 `start-instance` 예제에서는 지정된 인스턴스를 시작합니다.

```
aws lightsail start-instance \
    --instance-name WordPress-1
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조StartInstance의 섹션을 참조하세요. AWS CLI

### `start-relational-database`

다음 코드 예시에서는 `start-relational-database`을 사용하는 방법을 보여 줍니다.

AWS CLI

관계형 데이터베이스를 시작하려면

다음 `start-relational-database` 예제에서는 지정된 관계형 데이터베이스를 시작합니다.

```
aws lightsail start-relational-database \
```

```
    --relational-database-name Database-1
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조StartRelationalDatabase의 섹션을 참조하세요. AWS CLI

### stop-instance

다음 코드 예시에서는 stop-instance을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스를 중지하려면

다음 stop-instance 예제에서는 지정된 인스턴스를 중지합니다.

```
aws lightsail stop-instance \
--instance-name WordPress-1
```

출력:

```
{
```

```
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조StopInstance의 섹션을 참조하세요. AWS CLI

### stop-relational-database

다음 코드 예시에서는 stop-relational-database을 사용하는 방법을 보여 줍니다.

AWS CLI

관계형 데이터베이스를 중지하려면

다음 stop-relational-database 예제에서는 지정된 관계형 데이터베이스를 중지합니다.

```
aws lightsail stop-relational-database \
    --relational-database-name Database-1
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조StopRelationalDatabase의 섹션을 참조하세요. AWS CLI

### unpeer-vpc

다음 코드 예시에서는 unpeer-vpc을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon Lightsail 가상 프라이빗 클라우드를 피어링 해제하려면

다음 unpeer-vpc 예제에서는 지정된 AWS 리전에 대해 Amazon Lightsail 가상 프라이빗 클라우드(VPC)를 피어링합니다.

```
aws lightsail unpeer-vpc \
    --region us-west-2
```

출력:

```
{
    "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
    }
}
```

- 자세한 API 내용은 명령 참조UnpeerVpc의 섹션을 참조하세요. AWS CLI

## 를 사용한 Macie 예제 AWS CLI

다음 코드 예제에서는 Macie AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `describe-buckets`

다음 코드 예시에서는 `describe-buckets`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon Macie가 계정에 대해 모니터링하고 분석하는 하나 이상의 S3 버킷에 대한 데이터를 쿼리하려면

다음 `describe-buckets` 예제는 이름이 MY-S3로 시작되고 현재 AWS 리전에 있는 모든 S3 버킷에 대한 메타데이터를 쿼리합니다. MY-S3

```
aws macie2 describe-buckets \
    --criteria '{"bucketName":{"prefix":"my-S3"}}'
```

출력:


```
{
    "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
        }
    ]
}
```

자세한 내용은 Amazon Macie 사용 설명서의 S3 버킷 인벤토리 필터링을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeBuckets의 섹션을 참조하세요. AWS CLI

# 를 사용한 Amazon Managed Grafana 예제 AWS CLI

다음 코드 예제에서는 Amazon Managed Grafana와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### `list-workspaces`

다음 코드 예시에서는 `list-workspaces`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 자격 증명에 지정된 리전에서 계정의 워크스페이스를 나열하려면

다음 `list-workspaces` 예제에서는 계정의 리전에 대한 Grafana 워크스페이스를 나열합니다.

```
aws grafana list-workspaces
```

출력:


```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListWorkspaces의 섹션을 참조하세요. AWS CLI

## MediaConnect 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 MediaConnect.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



### 작업

#### `add-flow-outputs`

다음 코드 예시에서는 `add-flow-outputs`을 사용하는 방법을 보여 줍니다.

AWS CLI

흐름에 출력을 추가하려면

다음 `add-flow-outputs` 예제에서는 지정된 흐름에 출력을 추가합니다.

```
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
```

출력:

```
{
```

```
    "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:mediaconnect: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:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame"
}
```

자세한 내용은 AWS 요소 MediaConnect 사용 설명서의 흐름에 출력 추가를 참조하세요.

- 자세한 API 내용은 명령 참조AddFlowOutputs의 섹션을 참조하세요. AWS CLI

### `create-flow`

다음 코드 예시에서는 `create-flow`을 사용하는 방법을 보여 줍니다.

AWS CLI

흐름을 생성하려면

다음 `create-flow` 예제에서는 지정된 구성으로 흐름을 생성합니다.

```
aws mediaconnect 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
```

출력:

```
{
    "Flow": {
        "FlowArn": "arn:aws:mediaconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:ExampleFlow",
        "AvailabilityZone": "us-west-2c",
        "EgressIp": "54.245.71.21",
        "Source": {
            "IngestPort": 1055,
            "SourceArn": "arn:aws:mediaconnect: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"
    }
}
```

자세한 내용은 요소 사용 설명서의 흐름 생성을 참조하세요. AWS MediaConnect

- 자세한 API 내용은 명령 참조CreateFlow의 섹션을 참조하세요. AWS CLI

### `delete-flow`

다음 코드 예시에서는 `delete-flow`을 사용하는 방법을 보여 줍니다.

AWS CLI

흐름을 삭제하려면

다음 delete-flow 예제에서는 지정된 흐름을 삭제합니다.

```
aws mediaconnect delete-flow \
    --flow-arn arn:aws:mediaconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow
```

출력:

```
{
    "FlowArn": "arn:aws:mediaconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow",
    "Status": "DELETING"
}
```

자세한 내용은 AWS 요소 MediaConnect 사용 설명서의 흐름 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteFlow의 섹션을 참조하세요. AWS CLI

### describe-flow

다음 코드 예시에서는 describe-flow을 사용하는 방법을 보여 줍니다.

AWS CLI

흐름의 세부 정보를 보려면

다음 describe-flow 예제에서는 , 가용 영역, 상태ARN, 소스, 권한 및 출력과 같은 지정된 흐름의 세부 정보를 표시합니다.

```
aws mediaconnect describe-flow \
    --flow-arn arn:aws:mediaconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow
```

출력:

```
{
    "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:mediaconnect: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:mediaconnect:us-
east-1:123456789012:output:2-987655dEF67hiJ89-c34de5fG678h:LA",
                "Destination": "192.0.2.0"
            }
        ],
        "Source": {
            "IngestIp": "54.201.4.39",
```

```
            "SourceArn": "arn:aws:mediaconnect:us-
east-1:123456789012:source:3-4aBC56dEF78hiJ90-4de5fG6Hi78Jk:ShowSource",
            "Transport": {
                "MaxBitrate": 80000000,
                "Protocol": "rtp"
            },
            "IngestPort": 1069,
            "Description": "Saturday night show",
            "Name": "ShowSource",
            "WhitelistCidr": "10.24.34.0/23"
        }
    }
}
```

자세한 내용은 AWS 요소 MediaConnect 사용 설명서의 흐름 세부 정보 보기를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeFlow의 섹션을 참조하세요. AWS CLI

### grant-flow-entitlements

다음 코드 예시에서는 grant-flow-entitlements을 사용하는 방법을 보여 줍니다.

AWS CLI

흐름에 대한 권한을 부여하려면

다음 grant-flow-entitlements 예제에서는 지정된 기존 흐름에 다른 AWS 계정과 콘텐츠를 공유할 수 있는 권한을 부여합니다.

```
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
```

출력:

```
{
```

```
    "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:mediaconnect:us-
west-2:111122223333:entitlement:1-3333cccc4444dddd-1111aaaa2222:ExampleCorp",
            "Subscribers": [
                "777788889999"
            ],
            "Description": "For Example Corp"
        }
    ],
    "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame"
}
```

자세한 내용은 AWS 요소 MediaConnect 사용 설명서의 흐름에 대한 권한 부여를 참조하세요.

- 자세한 API 내용은 명령 참조GrantFlowEntitlements의 섹션을 참조하세요. AWS CLI

### `list-entitlements`

다음 코드 예시에서는 `list-entitlements`을 사용하는 방법을 보여 줍니다.

AWS CLI

권한 목록을 보려면

다음 `list-entitlements` 예제에서는 계정에 부여된 모든 권한 목록을 보여줍니다.

```
aws mediaconnect list-entitlements
```

출력:

```
{
    "Entitlements": [
        {
            "EntitlementArn": "arn:aws:mediaconnect:us-
west-2:111122223333:entitlement:1-11aa22bb11aa22bb-3333cccc4444:MyEntitlement",
            "EntitlementName": "MyEntitlement"
        }
    ]
}
```

자세한 내용은 요소 참조ListEntitlements의 섹션을 참조하세요. AWS MediaConnect API

- 자세한 API 내용은 명령 참조ListEntitlements의 섹션을 참조하세요. AWS CLI

### `list-flows`

다음 코드 예시에서는 `list-flows`을 사용하는 방법을 보여 줍니다.

AWS CLI

흐름 목록을 보려면

다음 `list-flows` 예제에서는 흐름 목록을 표시합니다.

```
aws mediaconnect list-flows
```

출력:

```
{
    "Flows": [
        {
            "Status": "STANDBY",
            "SourceType": "OWNED",
            "AvailabilityZone": "us-west-2a",
            "Description": "NYC awards show",
            "Name": "AwardsShow",
```

```
            "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow"
        },
        {
            "Status": "STANDBY",
            "SourceType": "OWNED",
            "AvailabilityZone": "us-west-2c",
            "Description": "LA basketball game",
            "Name": "BasketballGame",
            "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame"
        }
    ]
}
```

자세한 내용은 AWS 요소 MediaConnect 사용 설명서의 흐름 목록 보기를 참조하세요.

- 자세한 API 내용은 명령 참조ListFlows의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

MediaConnect 리소스의 태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 지정된 MediaConnect 리소스와 연결된 태그 키 및 값을 표시합니다.

```
aws mediaconnect list-tags-for-resource \
    --resource-arn arn:aws:mediaconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame
```

출력:

```
{
    "Tags": {
        "region": "west",
        "stage": "prod"
    }
}
```

자세한 내용은 AWS 요소 MediaConnect API 참조의 ListTagsForResource TagResource, UntagResource 섹션을 참조하세요.

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### `remove-flow-output`

다음 코드 예시에서는 `remove-flow-output`을 사용하는 방법을 보여 줍니다.

AWS CLI

흐름에서 출력을 제거하려면

다음 `remove-flow-output` 예제에서는 지정된 흐름에서 출력을 제거합니다.

```
aws mediaconnect remove-flow-output \
    --flow-arn arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame \
    --output-arn arn:aws:mediaconnect:us-
east-1:111122223333:output:2-3aBC45dEF67hiJ89-c34de5fG678h:NYC
```

출력:

```
{
    "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame",
    "OutputArn": "arn:aws:mediaconnect:us-
east-1:111122223333:output:2-3aBC45dEF67hiJ89-c34de5fG678h:NYC"
}
```

자세한 내용은 AWS 요소 MediaConnect 사용 설명서의 흐름에서 출력 제거를 참조하세요.

- 자세한 API 내용은 명령 참조RemoveFlowOutput의 섹션을 참조하세요. AWS CLI

### `revoke-flow-entitlement`

다음 코드 예시에서는 `revoke-flow-entitlement`을 사용하는 방법을 보여 줍니다.

AWS CLI

권한을 취소하려면

다음 `revoke-flow-entitlement` 예제에서는 지정된 흐름에 대한 권한을 취소합니다.

```
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
```

출력:

```
{
    "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"
}
```

자세한 내용은 요소 사용 설명서의 권한 취소를 참조하세요. AWS MediaConnect

- 자세한 API 내용은 명령 참조RevokeFlowEntitlement의 섹션을 참조하세요. AWS CLI

### `start-flow`

다음 코드 예시에서는 `start-flow`을 사용하는 방법을 보여 줍니다.

AWS CLI

흐름을 시작하려면

다음 `start-flow` 예제에서는 지정된 흐름을 시작합니다.

```
aws mediaconnect start-flow \
    --flow-arn arn:aws:mediaconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow
```

이 명령은 출력을 생성하지 않습니다. 출력:

```
{
    "FlowArn": "arn:aws:mediaconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow",
    "Status": "STARTING"
}
```

자세한 내용은 요소 사용 설명서의 흐름 시작을 참조하세요. AWS MediaConnect

- 자세한 API 내용은 명령 참조StartFlow의 섹션을 참조하세요. AWS CLI

### stop-flow

다음 코드 예시에서는 stop-flow을 사용하는 방법을 보여 줍니다.

AWS CLI

흐름을 중지하려면

다음 stop-flow 예제에서는 지정된 흐름을 중지합니다.

```
aws mediaconnect stop-flow \
    --flow-arn arn:aws:mediaconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow
```

출력:

```
{
    "Status": "STOPPING",
    "FlowArn": "arn:aws:mediaconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow"
}
```

자세한 내용은 요소 사용 설명서의 흐름 중지를 참조하세요. AWS MediaConnect

- 자세한 API 내용은 명령 참조StopFlow의 섹션을 참조하세요. AWS CLI

### tag-resource

다음 코드 예시에서는 tag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

MediaConnect 리소스에 태그를 추가하려면

다음 tag-resource 예제에서는 키 이름과 값이 있는 태그를 지정된 MediaConnect 리소스에 추가합니다.

```
aws mediaconnect tag-resource \
    --resource-arn arn:aws:mediaconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame
```

```
    --tags region=west
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS 요소 MediaConnect API 참조의 ListTagsForResource TagResource, UntagResource 섹션을 참조하세요.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### `untag-resource`

다음 코드 예시에서는 `untag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

MediaConnect 리소스에서 태그를 제거하려면

다음 `untag-resource` 예제에서는 지정된 키 이름과 관련 값이 있는 태그를 MediaConnect 리소스에서 제거합니다.

```
aws mediaconnect untag-resource \
    --resource-arn arn:aws:mediaconnect:us-
east-1:123456789012:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BasketballGame \
    --tag-keys region
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS 요소 MediaConnect API 참조의 ListTagsForResource TagResource, UntagResource 섹션을 참조하세요.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### `update-flow-entitlement`

다음 코드 예시에서는 `update-flow-entitlement`을 사용하는 방법을 보여 줍니다.

AWS CLI

권한을 업데이트하려면

다음 `update-flow-entitlement` 예제에서는 지정된 권한을 새 설명 및 구독자로 업데이트합니다.

```
aws mediaconnect update-flow-entitlement \
    --flow-arn arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame \
    --entitlement-arn arn:aws:mediaconnect:us-
west-2:111122223333:entitlement:1-11aa22bb11aa22bb-3333cccc4444:AnyCompany_Entitlement
 \
    --description 'For AnyCompany Affiliate' \
    --subscribers 777788889999
```

출력:

```
{
    "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:BaseballGame",
    "Entitlement": {
        "Name": "AnyCompany_Entitlement",
        "Description": "For AnyCompany Affiliate",
        "EntitlementArn": "arn:aws:mediaconnect:us-
west-2:111122223333:entitlement:1-11aa22bb11aa22bb-3333cccc4444:AnyCompany_Entitlement",
        "Encryption": {
            "KeyType": "static-key",
            "Algorithm": "aes128",
            "RoleArn": "arn:aws:iam::111122223333:role/MediaConnect-ASM",
            "SecretArn": "arn:aws:secretsmanager:us-
west-2:111122223333:secret:mySecret1"
        },
        "Subscribers": [
            "777788889999"
        ]
    }
}
```

자세한 내용은 AWS 요소 MediaConnect 사용 설명서의 권한 업데이트를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateFlowEntitlement의 섹션을 참조하세요. AWS CLI

### update-flow-output

다음 코드 예시에서는 update-flow-output을 사용하는 방법을 보여 줍니다.

AWS CLI

흐름의 출력을 업데이트하려면

다음 update-flow-output 예제에서는 지정된 흐름에 대한 출력을 업데이트합니다.

```
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
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS 요소 MediaConnect 사용 설명서의 흐름에 대한 출력 업데이트를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateFlowOutput의 섹션을 참조하세요. AWS CLI

### update-flow-source

다음 코드 예시에서는 update-flow-source을 사용하는 방법을 보여 줍니다.

AWS CLI

기존 흐름의 소스를 업데이트하려면

다음 update-flow-source 예제에서는 기존 흐름의 소스를 업데이트합니다.

```
aws mediaconnect update-flow-source \
```

```
    --flow-arn arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow \
    --source-arn arn:aws:mediaconnect:us-
east-1:111122223333:source:3-4aBC56dEF78hiJ90-4de5fG6Hi78Jk:ShowSource \
    --description 'Friday night show' \
    --ingest-port 3344 \
    --protocol rtp-fec \
    --whitelist-cidr 10.24.34.0/23
```

출력:

```
{
    "FlowArn": "arn:aws:mediaconnect:us-
east-1:111122223333:flow:1-23aBC45dEF67hiJ8-12AbC34DE5fG:AwardsShow",
    "Source": {
        "IngestIp": "34.210.136.56",
        "WhitelistCidr": "10.24.34.0/23",
        "Transport": {
            "Protocol": "rtp-fec"
        },
        "IngestPort": 3344,
        "Name": "ShowSource",
        "Description": "Friday night show",
        "SourceArn": "arn:aws:mediaconnect:us-
east-1:111122223333:source:3-4aBC56dEF78hiJ90-4de5fG6Hi78Jk:ShowSource"
    }
}
```

자세한 내용은 AWS 요소 MediaConnect 사용 설명서의 흐름 소스 업데이트를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateFlowSource의 섹션을 참조하세요. AWS CLI

## MediaConvert 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 MediaConvert.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### cancel-job

다음 코드 예시에서는 cancel-job을 사용하는 방법을 보여 줍니다.

AWS CLI

대기열에 있는 작업을 취소하려면

다음 cancel-job 예제에서는 ID 를 사용하여 작업을 취소합니다1234567891234-abc123. 서비스가 처리를 시작한 작업은 취소할 수 없습니다.

```
aws mediaconvert cancel-job \
    --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com \
    --region region-name-1 \
    --id 1234567891234-abc123
```

계정별 엔드포인트를 가져오려면 describe-endpoints를 사용하거나 엔드포인트 없이 명령을 전송하세요. 서비스가 오류와 엔드포인트를 반환합니다.

자세한 내용은 AWS 요소 사용 설명서의 요소 MediaConvert 작업 작업을 참조하세요. AWS MediaConvert

- 자세한 API 내용은 명령 참조CancelJob의 섹션을 참조하세요. AWS CLI

### create-job-template

다음 코드 예시에서는 create-job-template을 사용하는 방법을 보여 줍니다.

AWS CLI

작업 템플릿을 생성하는 방법

다음 create-job-template 예제에서는 job-template.json 시스템에 있는 파일에 지정된 트랜스코딩 설정을 사용하여 작업 템플릿을 생성합니다.

```
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
```

get-job-template 를 사용한 다음 JSON 파일을 수정하여 작업 템플릿 파일을 생성하는 경우 JobTemplate 객체를 제거하되 설정 하위 객체는 그 안에 유지합니다. 또한 , LastUpdated, 및 키-값 페어를 제거해야 합니다ArnTypeCreatedAt. JSON 파일 또는 명령줄에서 범주, 설명, 이름 및 대기열을 지정할 수 있습니다.

계정별 엔드포인트를 가져오려면 describe-endpoints를 사용하거나 엔드포인트 없이 명령을 전송하세요. 서비스가 오류와 엔드포인트를 반환합니다.

요청이 성공하면 서비스는 생성한 작업 템플릿의 JSON 사양을 반환합니다.

자세한 내용은 AWS 요소 사용 설명서의 요소 MediaConvert 작업 템플릿 작업을 참조하세요. AWS MediaConvert

- 자세한 API 내용은 명령 참조CreateJobTemplate의 섹션을 참조하세요. AWS CLI

### create-job

다음 코드 예시에서는 create-job을 사용하는 방법을 보여 줍니다.

AWS CLI

작업을 생성하는 방법

다음 create-job 예시에서는 명령을 보내는 소스 시스템에 있는 job.json 파일에 지정된 설정을 사용하여 트랜스코딩 작업을 생성합니다. 이 JSON 작업 사양은 각 설정을 개별적으로 지정하거나, 작업 템플릿을 참조하거나, 출력 사전 설정을 참조할 수 있습니다.

```
aws mediaconvert create-job \
    --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com \
    --region region-name-1 \
    --cli-input-json file://~/job.json
```

AWS Elemental MediaConvert 콘솔을 사용하여 JSON 작업 설정을 선택한 다음 작업 섹션 하단에서 작업 표시를 JSON 선택하여 작업 사양을 생성할 수 있습니다.

계정별 엔드포인트를 가져오려면 describe-endpoints를 사용하거나 엔드포인트 없이 명령을 전송하세요. 서비스가 오류와 엔드포인트를 반환합니다.

요청이 성공하면 서비스는 요청과 함께 전송한 JSON 작업 사양을 반환합니다.

자세한 내용은 AWS 요소 사용 설명서의 요소 MediaConvert 작업 작업을 참조하세요. AWS MediaConvert

- 자세한 API 내용은 명령 참조CreateJob의 섹션을 참조하세요. AWS CLI

### create-preset

다음 코드 예시에서는 create-preset을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 출력 사전 설정을 생성하려면

다음 create-preset 예제에서는 파일 에 지정된 출력 설정을 기반으로 사용자 지정 출력 사전 설정을 생성합니다preset.json. JSON 파일 또는 명령줄에서 범주, 설명 및 이름을 지정할 수 있습니다.

```
aws mediaconvert create-preset \
    --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com
    --region region-name-1 \
    --cli-input-json file://~/preset.json
```

get-preset 를 사용한 다음 출력 JSON 파일을 수정하여 프리셋 파일을 생성하는 경우 , LastUpdated, 및 키-값 페어를 제거해야 합니다ArnTypeCreatedAt.

계정별 엔드포인트를 가져오려면 describe-endpoints를 사용하거나 엔드포인트 없이 명령을 전송하세요. 서비스가 오류와 엔드포인트를 반환합니다.

자세한 내용은 AWS 요소 사용 설명서의 요소 MediaConvert 출력 사전 설정 작업을 참조하세요. AWS MediaConvert

- 자세한 API 내용은 명령 참조CreatePreset의 섹션을 참조하세요. AWS CLI

### create-queue

다음 코드 예시에서는 create-queue을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 대기열을 생성하려면

다음 create-queue 예제에서는 사용자 지정 트랜스코딩 대기열을 생성합니다.

```
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."
```

계정별 엔드포인트를 가져오려면 describe-endpoints를 사용하거나 엔드포인트 없이 명령을 전송하세요. 서비스가 오류와 엔드포인트를 반환합니다.

출력:

```
{
    "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."
    }
}
```

자세한 내용은 AWS 요소 사용 설명서의 요소 MediaConvert 대기열 작업을 참조하세요. AWS MediaConvert

- 자세한 API 내용은 명령 참조CreateQueue의 섹션을 참조하세요. AWS CLI

## delete-job-template

다음 코드 예시에서는 delete-job-template을 사용하는 방법을 보여 줍니다.

AWS CLI

작업 템플릿을 삭제하려면

다음 delete-job-template 예제에서는 지정된 사용자 지정 작업 템플릿을 삭제합니다.

```
aws mediaconvert delete-job-template \
    --name "DASH Streaming" \
    --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

이 명령은 출력을 생성하지 않습니다. 를 실행aws mediaconvert list-job-templates하여 템플릿이 삭제되었는지 확인합니다.

자세한 내용은 AWS 요소 사용 설명서의 요소 MediaConvert 작업 템플릿 작업을 참조하세요. AWS MediaConvert

- 자세한 API 내용은 명령 참조DeleteJobTemplate의 섹션을 참조하세요. AWS CLI

### delete-preset

다음 코드 예시에서는 delete-preset을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 온디맨드 대기열을 삭제하려면

다음 delete-preset 예제에서는 지정된 사용자 지정 사전 설정을 삭제합니다.

```
aws mediaconvert delete-preset \
    --name SimpleMP4 \
    --endpoint-url  https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

이 명령은 출력을 생성하지 않습니다. 를 실행aws mediaconvert list-presets하여 프리셋이 삭제되었는지 확인합니다.

자세한 내용은 AWS 요소 사용 설명서의 요소 MediaConvert 출력 사전 설정 작업을 참조하세요. AWS MediaConvert

- 자세한 API 내용은 명령 참조DeletePreset의 섹션을 참조하세요. AWS CLI

### delete-queue

다음 코드 예시에서는 delete-queue을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 온디맨드 대기열을 삭제하려면

다음 delete-queue 예제에서는 지정된 사용자 지정 온디맨드 대기열을 삭제합니다.

기본 대기열은 삭제할 수 없습니다. 활성 요금제가 있거나 처리되지 않은 작업이 포함된 예약 대기열은 삭제할 수 없습니다.

```
aws mediaconvert delete-queue \
    --name Customer1 \
    --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

이 명령은 출력을 생성하지 않습니다. 를 실행aws mediaconvert list-queues하여 대기열이 삭제되었는지 확인합니다.

자세한 내용은 AWS 요소 사용 설명서의 요소 MediaConvert 대기열 작업을 참조하세요. AWS MediaConvert

- 자세한 API 내용은 명령 참조DeleteQueue의 섹션을 참조하세요. AWS CLI

### describe-endpoints

다음 코드 예시에서는 describe-endpoints을 사용하는 방법을 보여 줍니다.

AWS CLI

계정별 엔드포인트를 가져오려면

다음 describe-endpoints 예제에서는 서비스에 다른 요청을 보내는 데 필요한 엔드포인트를 검색합니다.

```
aws mediaconvert describe-endpoints
```

출력:

```
{
    "Endpoints": [
        {
            "Url": "https://abcd1234.mediaconvert.region-name-1.amazonaws.com"
        }
    ]
}
```

자세한 내용은 요소 참조 의 MediaConvert 사용 시작하기API를 참조하세요. AWS MediaConvert API

- 자세한 API 내용은 명령 참조DescribeEndpoints의 섹션을 참조하세요. AWS CLI

### `get-job-template`

다음 코드 예시에서는 `get-job-template`을 사용하는 방법을 보여 줍니다.

AWS CLI

작업 템플릿에 대한 세부 정보를 가져오려면

다음 `get-job-template` 예제에서는 지정된 사용자 지정 작업 템플릿의 JSON정의를 보여줍니다.

```
aws mediaconvert get-job-template \
    --name "DASH Streaming" \
    --endpoint-url https://abcd1234.mediaconvert.us-east-1.amazonaws.com
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS 요소 사용 설명서의 요소 MediaConvert 작업 템플릿 작업을 참조하세요. AWS MediaConvert

- 자세한 API 내용은 명령 참조GetJobTemplate의 섹션을 참조하세요. AWS CLI

### get-job

다음 코드 예시에서는 get-job을 사용하는 방법을 보여 줍니다.

AWS CLI

특정 작업에 대한 세부 정보를 가져오는 방법

다음 예시에서는 ID가 1234567890987-1ab2c3인 작업에 대한 정보를 요청합니다. 이 예시에서는 오류로 종료되었습니다.

```
aws mediaconvert get-job \
    --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com \
    --region region-name-1 \
    --id 1234567890987-1ab2c3
```

계정별 엔드포인트를 가져오려면 describe-endpoints를 사용하거나 엔드포인트 없이 명령을 전송하세요. 서비스가 오류와 엔드포인트를 반환합니다.

요청이 성공하면 서비스는 다음과 같이 작업 설정, 반환된 오류 및 기타 작업 데이터를 포함한 작업 정보가 포함된 JSON 파일을 반환합니다.

```
{
    "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
```

```
    }
}
```

자세한 내용은 AWS 요소 사용 설명서의 요소 MediaConvert 작업 작업을 참조하세요. AWS MediaConvert

- 자세한 API 내용은 명령 참조GetJob의 섹션을 참조하세요. AWS CLI

### get-preset

다음 코드 예시에서는 get-preset을 사용하는 방법을 보여 줍니다.

AWS CLI

특정 사전 설정에 대한 세부 정보를 가져오려면

다음 get-preset 예제에서는 지정된 사용자 지정 프리셋의 JSON정의를 요청합니다.

```
aws mediaconvert get-preset \
    --name SimpleMP4 \
    --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

출력:

```
{
    "Preset": {
        "Description": "Creates basic MP4 file. No filtering or preproccessing.",
        "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
    }
}
```

자세한 내용은 AWS 요소 사용 설명서의 요소 MediaConvert 출력 사전 설정 작업을 참조하세요. AWS MediaConvert

- 자세한 API 내용은 명령 참조GetPreset의 섹션을 참조하세요. AWS CLI

### get-queue

다음 코드 예시에서는 get-queue을 사용하는 방법을 보여 줍니다.

AWS CLI

대기열에 대한 세부 정보를 가져오려면

다음 get-queue 예제에서는 지정된 사용자 지정 대기열의 세부 정보를 검색합니다.

```
aws mediaconvert get-queue \
    --name Customer1 \
    --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS 요소 사용 설명서의 요소 MediaConvert 대기열 작업을 참조하세요. AWS MediaConvert

- 자세한 API 내용은 명령 참조GetQueue의 섹션을 참조하세요. AWS CLI

### list-job-templates

다음 코드 예시에서는 list-job-templates을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 사용자 지정 작업 템플릿을 나열하려면

다음 list-job-templates 예제에서는 현재 리전의 모든 사용자 지정 작업 템플릿을 나열합니다. 시스템 작업 템플릿을 나열하려면 다음 예제를 참조하세요.

```
aws mediaconvert list-job-templates \
    --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

출력:

```
{
    "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
        }
    ]
}
```

예제 2: MediaConvert 시스템 작업 템플릿을 나열하려면

다음 `list-job-templates` 예제에서는 모든 시스템 작업 템플릿을 나열합니다.

```
aws mediaconvert list-job-templates \
    --endpoint-url https://abcd1234.mediaconvert.us-east-1.amazonaws.com \
    --list-by SYSTEM
```

출력:

```
{
    "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>...
    ]
}
```

자세한 내용은 AWS 요소 사용 설명서의 요소 MediaConvert 작업 템플릿 작업을 참조하세요. AWS MediaConvert

- 자세한 API 내용은 명령 참조ListJobTemplates의 섹션을 참조하세요. AWS CLI

### list-jobs

다음 코드 예시에서는 list-jobs을 사용하는 방법을 보여 줍니다.

AWS CLI

리전 내 모든 작업에 대한 세부 정보를 가져오는 방법

다음 예시에서는 지정된 리전의 모든 작업에 대한 정보를 요청합니다.

```
aws mediaconvert list-jobs \
    --endpoint-url https://abcd1234.mediaconvert.region-name-1.amazonaws.com \
    --region region-name-1
```

계정별 엔드포인트를 가져오려면 `describe-endpoints`를 사용하거나 엔드포인트 없이 명령을 전송하세요. 서비스가 오류와 엔드포인트를 반환합니다.

자세한 내용은 AWS 요소 사용 설명서의 요소 MediaConvert 작업 작업을 참조하세요. AWS MediaConvert

- 자세한 API 내용은 명령 참조ListJobs의 섹션을 참조하세요. AWS CLI

### `list-presets`

다음 코드 예시에서는 `list-presets`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 사용자 지정 출력 사전 설정을 나열하려면

다음 `list-presets` 예제에서는 사용자 지정 출력 사전 설정을 나열합니다. 시스템 사전 설정을 나열하려면 다음 예제를 참조하세요.

```
aws mediaconvert list-presets \
    --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

출력:

```
{
    "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 preproccessing."
        },
        {
            "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."
        }
    ]
}
```

예제 2: 시스템 출력 사전 설정을 나열하려면

다음 `list-presets` 예제에서는 사용 가능한 MediaConvert 시스템 사전 설정을 나열합니다. 사용자 지정 사전 설정을 나열하려면 이전 예제를 참조하세요.

```
aws mediaconvert list-presets \
    --list-by SYSTEM \
    --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

출력:

```
{
    "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>...
            }
        }
    ]
}
```

자세한 내용은 AWS 요소 사용 설명서의 요소 MediaConvert 출력 사전 설정 작업을 참조하세요. AWS MediaConvert

- 자세한 API 내용은 명령 참조ListPresets의 섹션을 참조하세요. AWS CLI

### `list-queues`

다음 코드 예시에서는 `list-queues`을 사용하는 방법을 보여 줍니다.

AWS CLI

대기열을 나열하려면

다음 `list-queues` 예제에서는 모든 MediaConvert 대기열을 나열합니다.

```
aws mediaconvert list-queues \
    --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 AWS 요소 사용 설명서의 요소 MediaConvert 대기열 작업을 참조하세요. AWS MediaConvert

- 자세한 API 내용은 명령 참조ListQueues의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

MediaConvert 대기열, 작업 템플릿 또는 출력 사전 설정의 태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 지정된 출력 사전 설정의 태그를 나열합니다.

```
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
```

출력:

```
{
    "ResourceTags": {
        "Tags": {
            "customer": "zippyVideo"
        },
        "Arn": "arn:aws:mediaconvert:us-west-2:123456789012:presets/SimpleMP4"
    }
}
```

자세한 내용은 AWS 요소 사용 설명서의 요소 MediaConvert 대기열, 작업 템플릿 및 출력 사전 설정 태깅을 참조하세요. AWS MediaConvert

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### update-job-template

다음 코드 예시에서는 `update-job-template`을 사용하는 방법을 보여 줍니다.

AWS CLI

작업 템플릿을 변경하려면

다음 `update-job-template` 예제에서는 지정된 사용자 지정 작업 템플릿의 JSON 정의를 제공된 파일의 JSON 정의로 바꿉니다.

aws mediaconvert update-job-template --name File1 --endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com --cli-input-json file://~/job-template-update.json

job-template-update.json의 콘텐츠:

```
{
    "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
}
```

요청으로 인해 오류가 발생하더라도 시스템에서 요청과 함께 전송하는 JSON페이로드를 반환합니다. 따라서 JSON 반환된 이 반드시 작업 템플릿의 새 정의는 아닙니다.

JSON 페이로드가 길 수 있으므로 위로 스크롤하여 오류 메시지를 확인해야 할 수 있습니다.

자세한 내용은 AWS 요소 사용 설명서의 요소 MediaConvert 작업 템플릿 작업을 참조하세요. AWS MediaConvert

- 자세한 API 내용은 명령 참조UpdateJobTemplate의 섹션을 참조하세요. AWS CLI

### update-preset

다음 코드 예시에서는 update-preset을 사용하는 방법을 보여 줍니다.

AWS CLI

사전 설정을 변경하려면

다음 update-preset 예제는 지정된 사전 설정에 대한 설명을 대체합니다.

```
aws mediaconvert update-preset \
--name Customer1 \
--description "New description text."
--endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

이 명령은 출력을 생성하지 않습니다. 출력:

```
{
    "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
    }
}
```

자세한 내용은 AWS 요소 사용 설명서의 요소 MediaConvert 출력 사전 설정 작업을 참조하세요. AWS MediaConvert

- 자세한 API 내용은 명령 참조UpdatePreset의 섹션을 참조하세요. AWS CLI

### update-queue

다음 코드 예시에서는 update-queue을 사용하는 방법을 보여 줍니다.

AWS CLI

대기열을 변경하려면

다음 update-queue 예제에서는 상태를 로 변경하여 지정된 대기열을 일시 중지합니다PAUSED.

```
aws mediaconvert update-queue \
--name Customer1 \
--status PAUSED
--endpoint-url https://abcd1234.mediaconvert.us-west-2.amazonaws.com
```

출력:

```
{
    "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"
```

```
    }
}
```

자세한 내용은 AWS 요소 사용 설명서의 요소 MediaConvert 대기열 작업을 참조하세요. AWS MediaConvert

- 자세한 API 내용은 명령 참조UpdateQueue의 섹션을 참조하세요. AWS CLI

## MediaLive 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 MediaLive.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `create-channel`

다음 코드 예시에서는 `create-channel`을 사용하는 방법을 보여 줍니다.

AWS CLI

채널을 생성하려면

다음 `create-channel` 예제에서는 지정하려는 파라미터가 포함된 JSON 파일을 전달하여 채널을 생성합니다.

이 예제의 채널은 비디오, 오디오 및 임베디드 캡션이 포함된 소스에 연결하는 HLS PULL 입력을 수집합니다. 채널은 Akamai 서버를 대상으로 하는 HLS 출력 그룹 하나를 생성합니다. 출력 그룹에는 두 개의 출력이 포함되어 있습니다. 하나는 H.265 비디오 및 AAC 오디오용이고 다른 하나는 웹 VTT 캡션용이며 영어로만 제공됩니다.

이 예제 채널JSON의 에는 HLS PULL 입력을 사용하고 Akamai를 대상으로 하는 HLS 출력 그룹을 생성하는 채널에 필요한 최소 파라미터가 포함되어 있습니다. 에는 다음과 같은 기본 섹션이 JSON 포함되어 있습니다.

InputAttachments: 오디오 소스 하나와 캡션 소스 하나를 지정합니다. 이 채널의 단일 출력 그룹에 대한 두 IP 주소(URLs)Destinations가 포함된 는 소스에서 찾은 첫 번째 비디오를 MediaLive 추출하는 비디오 선택기를 지정하지 않습니다. 이러한 주소에는 암호가 필요합니다.EncoderSettings에는 하위 섹션이 포함되어 있습니다. AudioDescriptions이 하위 섹션은 채널에 의 소스를 사용하고 AAC 형식으로 오디오를 InputAttachments생성하는 하나의 오디오 출력 자산이 포함되어 있습니다. CaptionDescriptions이 하위 섹션은 채널에 의 소스를 사용하고 웹VTT 형식으로 캡션을 InputAttachments생성하는 하나의 캡션 출력 자산이 포함되어 있습니다. VideoDescriptions이 하위 섹션은 채널에 지정된 해상도와 함께 비디오 출력 자산이 하나 포함되어 있습니다. OutputGroups이 하위 섹션은 출력 그룹을 지정합니다. 이 예제에는 이름이 인 그룹이 하나 있습니다Akamai. 연결은 HLS 를 사용하여 이루어집니다PUT. 출력 그룹에는 두 개의 출력이 포함됩니다. 출력 중 하나는 비디오 자산(명명Video_high)과 오디오 자산(명명)입니다Audio_EN. 한 가지 출력은 캡션 자산(이라는 이름)에 대한 것입니다WebVTT_EN.

이 예제에서는 일부 파라미터에 값이 없거나 중첩된 빈 파라미터가 포함되어 있습니다. 예를 들어 출력 OutputSettings 에는 빈 파라미터 M3u8Settings 로 끝나는 중첩된 파라미터가 여러 개 Video_and_audio 포함되어 있습니다. 이 파라미터를 포함해야 하지만 하나, 여러 개 또는 모든 하위 파라미터를 생략할 수 있습니다. 즉, 하위 파라미터가 기본값을 취하거나 null이 됩니다.

이 예제 채널에 적용되지만 이 파일에 지정되지 않은 모든 파라미터는 기본값을 취하거나 null로 설정되거나 에서 생성된 고유한 값을 취합니다 MediaLive.

```
aws medialive create-channel \
    --cli-input-json file://channel-in-hls-out-hls-akamai.json
```

channel-in-hls-out-hls-akamai.json의 콘텐츠:

```
{
    "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"
        }
    }
}
```

출력:

출력은 JSON 파일의 내용과 다음 값을 반복합니다. 모든 파라미터는 알파벳순으로 정렬됩니다.

ARN 채널용입니다. 의 마지막 부분은 고유한 채널 IDARN입니다.`EgressEndpoints` 는 PUSH 입력에만 사용되므로 이 예제 채널에서는 비어 있습니다. 적용하면 MediaLive 해당 콘텐츠의 주소가 푸시된 주소를 표시합니다.`OutputGroups`, `Outputs`. 여기에는 포함되지 않았지만 이 채널과 관련된 파라미터를 포함하여 출력 그룹 및 출력에 대한 모든 파라미터가 표시됩니다. 파라미터가 비어 있거나(아마도 이 채널 구성에서 파라미터 또는 기능이 비활성화되었음을 나타냄) 적용될 기본값을 표시할 수 있습니다.`LogLevel` 는 기본값()으로 설정되어 있습니다DISABLED.`Tags` 는 기본값(null)`PipelinesRunningCount`으로 설정되어 있고 채널의 현재 상태를 `State` 보여줍니다.

자세한 내용은 요소 사용 설명서의 Scratch에서 채널 생성을 참조하세요. AWS MediaLive

- 자세한 API 내용은 명령 참조CreateChannel의 섹션을 참조하세요. AWS CLI

**create-input**

다음 코드 예시에서는 create-input을 사용하는 방법을 보여 줍니다.

AWS CLI

입력을 생성하려면

다음 create-input 예제에서는 이 유형의 HLS PULL 입력에 적용되는 파라미터가 포함된 JSON 파일을 전달하여 입력을 생성합니다. 이 예제 입력JSON의 는 수집에서 중복성을 지원하기 위해 입력에 두 개의 소스(주소)를 지정합니다. 이러한 주소에는 암호가 필요합니다.

```
aws medialive create-input \
    --cli-input-json file://input-hls-pull-news.json
```

input-hls-pull-news.json의 콘텐츠:

```
{
    "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"
}
```

출력:

출력은 JSON 파일의 내용과 다음 값을 반복합니다. 모든 파라미터는 알파벳순으로 정렬됩니다.

Arn 입력에 사용합니다. 의 마지막 부분은 새로 생성된 입력에 대해 항상 비어 Attached Channels있는 고유한 입력 IDARN입니다. Destinations는 PUSH 입력에만 사용되므로 이

예제에서는 비어 있습니다.Id 는 입력에 대해서만 사용MediaConnectFlows되므로 의 ID와 동일합니다ARN. 는 유형 의 입력에만 사용되므로 이 예제에서는 비어 SecurityGroups있습니다 MediaConnect. 는 PUSH 입력에 대해서만 사용되므로 비어 있습니다.State Tags는 비어 있습니다(이 파라미터의 기본값).

자세한 내용은 AWS 요소 MediaLive 사용 설명서의 입력 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateInput의 섹션을 참조하세요. AWS CLI

## MediaPackage 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 MediaPackage.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `create-channel`

다음 코드 예시에서는 `create-channel`을 사용하는 방법을 보여 줍니다.

AWS CLI

채널을 생성하려면

다음 `create-channel` 명령은 `sportschannel` 현재 계정에 라는 채널을 생성합니다.

```
aws mediapackage create-channel --id sportschannel
```

출력:

```
{
   "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"
    }
}
```

자세한 내용은 요소 사용 설명서의 채널 생성을 참조하세요. AWS MediaPackage

- 자세한 API 내용은 명령 참조CreateChannel의 섹션을 참조하세요. AWS CLI

### `create-origin-endpoint`

다음 코드 예시에서는 `create-origin-endpoint`을 사용하는 방법을 보여 줍니다.

AWS CLI

오리진 엔드포인트를 생성하려면

다음 `create-origin-endpoint` 명령은 JSON 파일에 제공된 패키지 설정과 지정된 엔드포인트 설정cmafsports으로 라는 오리진 엔드포인트를 생성합니다.

```
aws mediapackage create-origin-endpoint \
    --channel-id sportschannel \
    --id cmafsports \
    --cmaf-package file://file/path/cmafpkg.json --description "cmaf output of
 sports" \
    --id cmaf_sports \
    --manifest-name sports_channel \
    --startover-window-seconds 300 \
    --tags region=west,media=sports \
    --time-delay-seconds 10
```

출력:

```
{
    "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": []
}
```

자세한 내용은 AWS 요소 MediaPackage 사용 설명서의 엔드포인트 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateOriginEndpoint의 섹션을 참조하세요. AWS CLI

### `delete-channel`

다음 코드 예시에서는 `delete-channel`을 사용하는 방법을 보여 줍니다.

AWS CLI

채널을 삭제하려면

다음 `delete-channel` 명령은 라는 채널을 삭제합니다`test`.

```
aws mediapackage delete-channel \
    --id test
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS 요소 MediaPackage 사용 설명서의 채널 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteChannel의 섹션을 참조하세요. AWS CLI

### `delete-origin-endpoint`

다음 코드 예시에서는 `delete-origin-endpoint`을 사용하는 방법을 보여 줍니다.

AWS CLI

오리진 엔드포인트를 삭제하려면

다음 `delete-origin-endpoint` 명령은 라는 오리진 엔드포인트를 삭제합니다`tester2`.

```
aws mediapackage delete-origin-endpoint \
    --id tester2
```

자세한 내용은 AWS 요소 MediaPackage 사용 설명서의 엔드포인트 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteOriginEndpoint의 섹션을 참조하세요. AWS CLI

### `describe-channel`

다음 코드 예시에서는 `describe-channel`을 사용하는 방법을 보여 줍니다.

AWS CLI

채널을 설명하려면

다음 `describe-channel` 명령은 라는 채널의 모든 세부 정보를 표시합니다`test`.

```
aws mediapackage describe-channel \
    --id test
```

출력:

```
{
    "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": {}
```

```
}
```

자세한 내용은 AWS Elemental MediaPackage 사용 설명서의 채널 세부 정보 보기<https://docs.aws.amazon.com/mediapackage/latest/ug/channels-view.html>를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeChannel의 섹션을 참조하세요. AWS CLI

### `describe-origin-endpoint`

다음 코드 예시에서는 `describe-origin-endpoint`을 사용하는 방법을 보여 줍니다.

AWS CLI

오리진 엔드포인트를 설명하려면

다음 `describe-origin-endpoint` 명령은 라는 오리진 엔드포인트의 모든 세부 정보를 표시합니다`cmaf_sports`.

```
aws mediapackage describe-origin-endpoint \
    --id cmaf_sports
```

출력:

```
{
    "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": []
}
```

자세한 내용은 요소 사용 설명서의 단일 엔드포인트 보기를 참조하세요. AWS MediaPackage

- 자세한 API 내용은 명령 참조DescribeOriginEndpoint의 섹션을 참조하세요. AWS CLI

### `list-channels`

다음 코드 예시에서는 `list-channels`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 채널을 나열하는 방법

다음 `list-channels` 명령은 현재 AWS 계정에 구성된 모든 채널을 나열합니다.

```
aws mediapackage list-channels
```

출력:

```
{
    "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": {}
        }
    ]
}
```

자세한 내용은 요소 사용 설명서의 채널 세부 정보 보기를 참조하세요. AWS MediaPackage

- 자세한 API 내용은 명령 참조ListChannels의 섹션을 참조하세요. AWS CLI

### `list-origin-endpoints`

다음 코드 예시에서는 `list-origin-endpoints`을 사용하는 방법을 보여 줍니다.

AWS CLI

채널의 모든 오리진-엔드포인트를 나열하는 방법

다음 `list-origin-endpoints` 명령은 `test` 채널에 구성된 모든 오리진 엔드포인트를 나열합니다.

```
aws mediapackage list-origin-endpoints \
    --channel-id test
```

출력:

```
{
    "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": []
        }
    ]
}
```

자세한 내용은 AWS 요소 MediaPackage 사용 설명서의 채널과 연결된 모든 엔드포인트 보기를 참조하세요.

- 자세한 API 내용은 명령 참조ListOriginEndpoints의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 할당된 태그를 나열하려면

다음 `list-tags-for-resource` 명령은 지정된 리소스에 할당된 태그를 나열합니다.

```
aws mediapackage list-tags-for-resource \
    --resource-arn arn:aws:mediapackage:us-
west-2:111222333:channels/6d345804ec3f46c9b454a91d4a80d0e0
```

출력:

```
{
    "Tags": {
        "region": "west"
    }
}
```

자세한 내용은 Elemental 사용 설명서의 AWS Elemental에서 리소스 태그 지정 MediaPackage을 참조하세요. AWS MediaPackage

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### `rotate-ingest-endpoint-credentials`

다음 코드 예시에서는 `rotate-ingest-endpoint-credentials`을 사용하는 방법을 보여 줍니다.

AWS CLI

수집 보안 인증 정보를 교체하려면

다음 `rotate-ingest-endpoint-credentials` 명령은 지정된 수집 엔드포인트의 웹DAV 사용자 이름과 암호를 교대로 사용합니다.

```
aws mediapackage rotate-ingest-endpoint-credentials \
    --id test \
    --ingest-endpoint-id 584797f1740548c389a273585dd22a63
```

출력:

```
{
    "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": {}
}
```

자세한 내용은 요소 사용 설명서의 입력에 대한 자격 증명 교체URL를 참조하세요. AWS MediaPackage

- 자세한 API 내용은 명령 참조RotateIngestEndpointCredentials의 섹션을 참조하세요. AWS CLI

### `tag-resource`

다음 코드 예시에서는 `tag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 태그를 추가하려면

다음 `tag-resource` 명령은 `region=west` 키와 값 페어를 지정된 리소스에 추가합니다.

```
aws mediapackage tag-resource \
    --resource-arn arn:aws:mediapackage:us-
west-2:111222333:channels/6d345804ec3f46c9b454a91d4a80d0e0 \
    --tags region=west
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Elemental 사용 설명서의 AWS Elemental에서 리소스 태그 지정 MediaPackage을 참조하세요. AWS MediaPackage

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### `untag-resource`

다음 코드 예시에서는 `untag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에서 태그를 제거하려면

다음 `untag-resource` 명령은 지정된 채널`region`에서 키가 있는 태그를 제거합니다.

```
aws mediapackage untag-resource \
    --resource-arn arn:aws:mediapackage:us-
west-2:111222333:channels/6d345804ec3f46c9b454a91d4a80d0e0 \
    --tag-keys region
```

자세한 내용은 Elemental 사용 설명서의 AWS Elemental에서 리소스 태그 지정 MediaPackage을 참조하세요. AWS MediaPackage

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-channel

다음 코드 예시에서는 update-channel을 사용하는 방법을 보여 줍니다.

AWS CLI

채널을 업데이트하려면

다음 update-channel 명령은 라는 채널을 업데이트sportschannel하여 설명을 포함합니다24x7 sports.

```
aws mediapackage update-channel \
    --id sportschannel \
    --description "24x7 sports"
```

출력:

```
{
    "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": {}
}
```

자세한 내용은 AWS 요소 MediaPackage 사용 설명서의 채널 편집을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateChannel의 섹션을 참조하세요. AWS CLI

### update-origin-endpoint

다음 코드 예시에서는 update-origin-endpoint을 사용하는 방법을 보여 줍니다.

AWS CLI

오리진 엔드포인트를 업데이트하려면

다음 update-origin-endpoint 명령은 라는 오리진 엔드포인트를 업데이트합니다cmaf_sports. 시간 지연을 0 초로 변경합니다.

```
aws mediapackage update-origin-endpoint \
    --id cmaf_sports \
    --time-delay-seconds 0
```

출력:

```
{
    "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": []
}
```

자세한 내용은 AWS 요소 MediaPackage 사용 설명서의 엔드포인트 편집을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateOriginEndpoint의 섹션을 참조하세요. AWS CLI

## MediaPackage VOD 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 MediaPackage VOD.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### `create-asset`

다음 코드 예시에서는 `create-asset`을 사용하는 방법을 보여 줍니다.

AWS CLI

자산을 생성하려면

다음 `create-asset` 예제에서는 `Chicken_Asset` 현재 AWS 계정에 이름이 지정된 자산을 생성합니다. 자산은 파일을 `30sec_chicken.smil`에 수집합니다 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
```

출력:

```
{
    "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"
        }
    ]
```

```
}
```

자세한 내용은 요소 사용 설명서의 자산 삽입을 참조하세요. AWS MediaPackage

- 자세한 API 내용은 명령 참조CreateAsset의 섹션을 참조하세요. AWS CLI

### `create-packaging-configuration`

다음 코드 예시에서는 `create-packaging-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

패키징 구성을 생성하려면

다음 `create-packaging-configuration` 예제에서는 라는 패키징 그룹에 라는 `new_hls` 패키징 구성을 생성합니다`hls_chicken`. 이 예제에서는 라는 디스크의 파일을 사용하여 세부 정보를 `hls_pc.json` 제공합니다.

```
aws mediapackage-vod create-packaging-configuration \
    --id new_hls \
    --packaging-group-id hls_chicken \
    --hls-package file://hls_pc.json
```

`hls_pc.json`의 콘텐츠:

```
{
    "HlsManifests":[
        {
            "AdMarkers":"NONE",
            "IncludeIframeOnlyStream":false,
            "ManifestName":"string",
            "ProgramDateTimeIntervalSeconds":60,
            "RepeatExtXKey":true,
            "StreamSelection":{
                "MaxVideoBitsPerSecond":1000,
                "MinVideoBitsPerSecond":0,
                "StreamOrder":"ORIGINAL"
            }
        }
    ],
    "SegmentDurationSeconds":6,
    "UseAudioRenditionGroup":false
```

```
}
```

출력:

```
{
    "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"
                }
            }
        ]
    }
}
```

자세한 내용은 요소 사용 설명서의 패키징 구성 생성을 참조하세요. AWS MediaPackage

- 자세한 API 내용은 명령 참조CreatePackagingConfiguration의 섹션을 참조하세요. AWS CLI

### `create-packaging-group`

다음 코드 예시에서는 `create-packaging-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

패키징 그룹을 생성하려면

다음 `create-packaging-group` 예제에서는 현재 AWS 계정에 구성된 모든 패키징 그룹을 나열합니다.

```
aws mediapackage-vod create-packaging-group \
    --id hls_chicken
```

출력:

```
{
    "Arn": "arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-groups/
hls_chicken",
    "Id": "hls_chicken"
}
```

자세한 내용은 요소 사용 설명서의 패키징 그룹 생성을 참조하세요. AWS MediaPackage

- 자세한 API 내용은 명령 참조CreatePackagingGroup의 섹션을 참조하세요. AWS CLI

### delete-asset

다음 코드 예시에서는 delete-asset을 사용하는 방법을 보여 줍니다.

AWS CLI

자산을 삭제하려면

다음 delete-asset 예제에서는 라는 자산을 삭제합니다30sec_chicken.

```
aws mediapackage-vod delete-asset \
    --id 30sec_chicken
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS 요소 MediaPackage 사용 설명서의 자산 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteAsset의 섹션을 참조하세요. AWS CLI

### delete-packaging-configuration

다음 코드 예시에서는 delete-packaging-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

패키징 구성을 삭제하려면

다음 delete-packaging-configuration 예제에서는 라는 패키징 구성을 삭제합니다CMAF.

```
aws mediapackage-vod delete-packaging-configuration \
    --id CMAF
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS 요소 MediaPackage 사용 설명서의 패키징 구성 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeletePackagingConfiguration의 섹션을 참조하세요. AWS CLI

### delete-packaging-group

다음 코드 예시에서는 delete-packaging-group을 사용하는 방법을 보여 줍니다.

AWS CLI

패키징 그룹을 삭제하려면

다음 delete-packaging-group 예제에서는 라는 패키징 그룹을 삭제합니다Dash_widevine.

```
aws mediapackage-vod delete-packaging-group \
    --id Dash_widevine
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS 요소 MediaPackage 사용 설명서의 패키징 그룹 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeletePackagingGroup의 섹션을 참조하세요. AWS CLI

### describe-asset

다음 코드 예시에서는 describe-asset을 사용하는 방법을 보여 줍니다.

AWS CLI

자산을 설명하려면

다음 describe-asset 예제에서는 라는 자산의 모든 세부 정보를 표시합니다30sec_chicken.

```
aws mediapackage-vod describe-asset \
```

```
    --id 30sec_chicken
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 요소 사용 설명서의 자산 세부 정보 보기를 참조하세요. AWS MediaPackage

- 자세한 API 내용은 명령 참조DescribeAsset의 섹션을 참조하세요. AWS CLI

### describe-packaging-configuration

다음 코드 예시에서는 describe-packaging-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

패키징 구성을 설명하려면

다음 describe-packaging-configuration 예제에서는 라는 패키징 구성의 모든 세부 정보를 표시합니다DASH.

```
aws mediapackage-vod describe-packaging-configuration \
    --id DASH
```

출력:

```
{
    "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"
            }
        }
    ]
}
```

자세한 내용은 요소 사용 설명서의 패키징 구성 세부 정보 보기를 참조하세요. AWS MediaPackage

- 자세한 API 내용은 명령 참조DescribePackagingConfiguration의 섹션을 참조하세요. AWS CLI

### describe-packaging-group

다음 코드 예시에서는 describe-packaging-group을 사용하는 방법을 보여 줍니다.

AWS CLI

패키징 그룹을 설명하려면

다음 describe-packaging-group 예제에서는 라는 패키징 그룹의 모든 세부 정보를 표시합니다Packaging_group_1.

```
aws mediapackage-vod describe-packaging-group \
    --id Packaging_group_1
```

출력:

```
{
    "Arn": "arn:aws:mediapackage-vod:us-west-2:111122223333:packaging-groups/
Packaging_group_1",
    "Id": "Packaging_group_1"
}
```

자세한 내용은 요소 사용 설명서의 패키징 그룹 세부 정보 보기를 참조하세요. AWS MediaPackage

- 자세한 API 내용은 명령 참조DescribePackagingGroup의 섹션을 참조하세요. AWS CLI

### list-assets

다음 코드 예시에서는 list-assets을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 자산을 나열하려면

다음 list-assets 예제에서는 현재 AWS 계정에 구성된 모든 자산을 나열합니다.

```
aws mediapackage-vod list-assets
```

출력:

```
{
    "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"
    ]
}
```

자세한 내용은 AWS 요소 MediaPackage 사용 설명서의 자산 세부 정보 보기를 참조하세요.

- 자세한 API 내용은 명령 참조ListAssets의 섹션을 참조하세요. AWS CLI

### `list-packaging-configurations`

다음 코드 예시에서는 `list-packaging-configurations`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 패키징 구성을 나열하려면

다음 `list-packaging-configurations` 예제에서는 라는 패키징 그룹에 구성된 모든 패키징 구성을 나열합니다`Packaging_group_1`.

```
aws mediapackage-vod list-packaging-configurations \
    --packaging-group-id Packaging_group_1
```

출력:

```
{
    "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"
                    }
                }
            ]
        }
    ]
}
```

자세한 내용은 요소 사용 설명서의 패키징 구성 세부 정보 보기를 참조하세요. AWS MediaPackage

- 자세한 API 내용은 명령 참조ListPackagingConfigurations의 섹션을 참조하세요. AWS CLI

### `list-packaging-groups`

다음 코드 예시에서는 `list-packaging-groups`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 패키징 그룹을 나열하려면

다음 `list-packaging-groups` 예제에서는 현재 AWS 계정에 구성된 모든 패키징 그룹을 나열합니다.

```
aws mediapackage-vod list-packaging-groups
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 요소 사용 설명서의 패키징 그룹 세부 정보 보기를 참조하세요. AWS MediaPackage

- 자세한 API 내용은 명령 참조ListPackagingGroups의 섹션을 참조하세요. AWS CLI

## MediaStore 를 사용한 데이터 플레인 예제 AWS CLI

다음 코드 예제에서는 MediaStore Data Plane과 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### delete-object

다음 코드 예시에서는 delete-object을 사용하는 방법을 보여 줍니다.

AWS CLI

객체를 삭제하려면

다음 delete-object 예제에서는 지정된 객체를 삭제합니다.

```
aws mediastore-data delete-object \
    --endpoint=https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com \
    --path=/folder_name/README.md
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS 요소 MediaStore 사용 설명서의 객체 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteObject의 섹션을 참조하세요. AWS CLI

### describe-object

다음 코드 예시에서는 describe-object을 사용하는 방법을 보여 줍니다.

AWS CLI

객체의 헤더를 보려면

다음 describe-object 예제에서는 지정된 경로에 있는 객체의 헤더를 표시합니다.

```
aws mediastore-data describe-object \
    --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com \
    --path events/baseball/setup.jpg
```

출력:

```
{
```

```
    "LastModified": "Fri, 19 Jul 2019 21:50:31 GMT",
    "ContentType": "image/jpeg",
    "ContentLength": "3860266",
    "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3"
}
```

자세한 내용은 AWS 요소 MediaStore 사용 설명서의 객체 세부 정보 보기를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeObject의 섹션을 참조하세요. AWS CLI

### get-object

다음 코드 예시에서는 get-object을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 전체 객체 다운로드

다음 get-object 예제에서는 지정된 객체를 다운로드합니다.

```
aws mediastore-data get-object \
    --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com \
    --path events/baseball/setup.jpg setup.jpg
```

출력:

```
{
    "ContentType": "image/jpeg",
    "StatusCode": 200,
    "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3",
    "ContentLength": "3860266",
    "LastModified": "Fri, 19 Jul 2019 21:50:31 GMT"
}
```

예제 2: 객체의 일부를 다운로드하는 방법

다음 get-object 예제에서는 객체의 지정된 부분을 다운로드합니다.

```
aws mediastore-data get-object \
    --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com \
    --path events/baseball/setup.jpg setup.jpg \
    --range "bytes=0-100"
```

출력:

```
{
    "StatusCode": 206,
    "LastModified": "Fri, 19 Jul 2019 21:50:31 GMT",
    "ContentType": "image/jpeg",
    "ContentRange": "bytes 0-100/3860266",
    "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3",
    "ContentLength": "101"
}
```

자세한 내용은 AWS 요소 MediaStore 사용 설명서의 객체 다운로드를 참조하세요.

- 자세한 API 내용은 명령 참조GetObject의 섹션을 참조하세요. AWS CLI

### `list-items`

다음 코드 예시에서는 `list-items`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 컨테이너에 저장된 항목(객체 및 폴더) 목록을 보려면

다음 `list-items` 예제에서는 지정된 컨테이너에 저장된 항목(객체 및 폴더) 목록을 표시합니다.

```
aws mediastore-data list-items \
    --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com
```

출력:

```
{
    "Items": [
        {
            "Type": "OBJECT",
            "ContentLength": 3784,
            "Name": "setup.jpg",
            "ETag":
 "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3",
            "ContentType": "image/jpeg",
            "LastModified": 1563571859.379
        },
        {
```

```
            "Type": "FOLDER",
            "Name": "events"
        }
    ]
}
```

예제 2: 폴더에 저장된 항목(객체 및 폴더) 목록을 보려면

다음 `list-items` 예제에서는 지정된 폴더에 저장된 항목(객체 및 폴더) 목록을 표시합니다.

```
aws mediastore-data list-items \
    --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com \
    --path events/baseball
```

출력:

```
{
    "Items": [
        {
            "ETag":
 "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3",
            "ContentType": "image/jpeg",
            "Type": "OBJECT",
            "ContentLength": 3860266,
            "LastModified": 1563573031.872,
            "Name": "setup.jpg"
        }
    ]
}
```

자세한 내용은 AWS 요소 MediaStore 사용 설명서의 객체 목록 보기를 참조하세요.

- 자세한 API 내용은 명령 참조ListItems의 섹션을 참조하세요. AWS CLI

### put-object

다음 코드 예시에서는 `put-object`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 컨테이너에 객체 업로드

다음 put-object 예제에서는 지정된 컨테이너에 객체를 업로드합니다.

```
aws mediastore-data put-object \
    --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com \
    --body ReadMe.md \
    --path ReadMe.md \
    --cache-control "max-age=6, public" \
    --content-type binary/octet-stream
```

출력:

```
{
    "ContentSHA256":
 "f29bc64a9d3732b4b9035125fdb3285f5b6455778edca72414671e0ca3b2e0de",
    "StorageClass": "TEMPORAL",
    "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3"
}
```

예제 2: 컨테이너 내의 폴더에 객체 업로드

다음 put-object 예제에서는 컨테이너 내의 지정된 폴더에 객체를 업로드합니다.

```
aws mediastore-data put-object \
    --endpoint https://aaabbbcccdddee.data.mediastore.us-west-2.amazonaws.com \
    --body ReadMe.md \
    --path /september-events/ReadMe.md \
    --cache-control "max-age=6, public" \
    --content-type binary/octet-stream
```

출력:

```
{
    "ETag": "2aa333bbcc8d8d22d777e999c88d4aa9eeeeee4dd89ff7f555555555555da6d3",
    "ContentSHA256":
 "f29bc64a9d3732b4b9035125fdb3285f5b6455778edca72414671e0ca3b2e0de",
    "StorageClass": "TEMPORAL"
}
```

자세한 내용은 AWS 요소 MediaStore 사용 설명서의 객체 업로드를 참조하세요.

- 자세한 API 내용은 명령 참조PutObject의 섹션을 참조하세요. AWS CLI

# MediaTailor 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 MediaTailor.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `delete-playback-configuration`

다음 코드 예시에서는 `delete-playback-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

구성을 삭제하려면

다음은 라는 구성을 `delete-playback-configuration` 삭제합니다`campaign_short`.

```
aws mediatailor delete-playback-configuration \
    --name campaign_short
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS 요소 MediaTailor 사용 설명서의 구성 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeletePlaybackConfiguration의 섹션을 참조하세요. AWS CLI

### `get-playback-configuration`

다음 코드 예시에서는 `get-playback-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

구성을 설명하려면

다음은 라는 구성의 모든 세부 정보를 get-playback-configuration 표시합니다west_campaign.

```
aws mediatailor get-playback-configuration \
    --name west_campaign
```

출력:

```
{
     "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"
}
```

자세한 내용은 AWS 요소 MediaTailor 사용 설명서의 구성 보기를 참조하세요.

- 자세한 API 내용은 명령 참조GetPlaybackConfiguration의 섹션을 참조하세요. AWS CLI

### list-playback-configurations

다음 코드 예시에서는 list-playback-configurations을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 구성을 나열하려면

다음은 현재 AWS 계정의 구성에 대한 모든 세부 정보를 list-playback-configurations 표시합니다.

```
aws mediatailor list-playback-configurations
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 요소 사용 설명서의 구성 보기<https://docs.aws.amazon.com/mediatailor/latest/ug/configurations-view.html>를 참조하세요. AWS MediaTailor

- 자세한 API 내용은 명령 참조ListPlaybackConfigurations의 섹션을 참조하세요. AWS CLI

### `put-playback-configuration`

다음 코드 예시에서는 `put-playback-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

구성을 생성하려면

다음은 라는 구성을 put-playback-configuration 생성합니다campaign_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
```

출력:

```
{
    "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"
}
```

자세한 내용은 요소 사용 설명서의 구성 생성을 참조하세요. AWS MediaTailor

- 자세한 API 내용은 명령 참조PutPlaybackConfiguration의 섹션을 참조하세요. AWS CLI

# 를 사용한 MemoryDB 예제 AWS CLI

다음 코드 예제에서는 MemoryDB와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `copy-snapshot`

다음 코드 예시에서는 copy-snapshot을 사용하는 방법을 보여 줍니다.

AWS CLI

스냅샷을 복사하려면

다음 copy-snapshot 예제에서는 스냅샷 사본을 생성합니다.

```
aws memorydb copy-snapshot \
    --source-snapshot-name my-cluster-snapshot \
    --target-snapshot-name my-cluster-snapshot-copy
```

출력

```
{
    "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
        }
    }
}
```

자세한 내용은 MemoryDB 사용 설명서의 스냅샷 복사를 참조하세요.

- 자세한 API 내용은 명령 참조CopySnapshot의 섹션을 참조하세요. AWS CLI

### `create-acl`

다음 코드 예시에서는 `create-acl`을 사용하는 방법을 보여 줍니다.

AWS CLI

를 생성하려면 ACL

다음 `create-acl` 예제에서는 새 액세스 제어 목록을 생성합니다.

```
aws memorydb create-acl \
    --acl-name "new-acl-1" \
    --user-names "my-user"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 MemoryDB 사용 설명서의 액세스 제어 목록을 사용하여 사용자 인증을 참조하세요.

- 자세한 API 내용은 명령 참조CreateAcl의 섹션을 참조하세요. AWS CLI

### `create-cluster`

다음 코드 예시에서는 `create-cluster`을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터를 생성하려면

다음 `create-cluster` 예제에서는 새 클러스터를 생성합니다.

```
aws memorydb create-cluster \
    --cluster-name my-new-cluster \
    --node-type db.r6g.large \
    --acl-name my-acl \
    --subnet-group my-sg
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 MemoryDB 사용 설명서의 클러스터 관리를 참조하세요.

- 자세한 API 내용은 명령 참조CreateCluster의 섹션을 참조하세요. AWS CLI

### `create-parameter-group`

다음 코드 예시에서는 `create-parameter-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

파라미터 그룹을 생성하려면

다음 `create-parameter-group` 예제에서는 파라미터 그룹을 생성합니다.

```
aws memorydb create-parameter-group \
    --parameter-group-name myRedis6x \
    --family memorydb_redis6 \
    --description "my-parameter-group"
```

출력:

```
{
    "ParameterGroup": {
        "Name": "myredis6x",
        "Family": "memorydb_redis6",
        "Description": "my-parameter-group",
        "ARN": "arn:aws:memorydb:us-east-1:49165xxxxxx:parametergroup/myredis6x"
    }
}
```

자세한 내용은 MemoryDB 사용 설명서의 파라미터 그룹 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateParameterGroup의 섹션을 참조하세요. AWS CLI

### create-snapshot

다음 코드 예시에서는 create-snapshot을 사용하는 방법을 보여 줍니다.

AWS CLI

스냅샷을 생성하려면

다음 create-snapshot 예제에서는 스냅샷을 생성합니다.

```
aws memorydb create-snapshot \
    --cluster-name my-cluster \
    --snapshot-name my-cluster-snapshot
```

출력:

```
{
    "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
        }
    }
}
```

자세한 내용은 MemoryDB 사용 설명서의 수동 스냅샷 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateSnapshot의 섹션을 참조하세요. AWS CLI

### create-subnet-group

다음 코드 예시에서는 create-subnet-group을 사용하는 방법을 보여 줍니다.

AWS CLI

서브넷 그룹을 생성하려면

다음 create-subnet-group 예제에서는 서브넷 그룹을 생성합니다.

```
aws memorydb create-subnet-group \
    --subnet-group-name mysubnetgroup \
    --description "my subnet group" \
    --subnet-ids subnet-5623xxxx
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 MemoryDB 사용 설명서의 서브넷 그룹 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateSubnetGroup의 섹션을 참조하세요. AWS CLI

### create-user

다음 코드 예시에서는 create-user을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자를 생성하려면

다음 create-user 예제에서는 새 사용자를 생성합니다.

```
aws memorydb create-user \
    --user-name user-name-1 \
    --access-string "~objects:* ~items:* ~public:*" \
     --authentication-mode \
         Passwords="enterapasswordhere",Type=password
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 MemoryDB 사용 설명서의 액세스 제어 목록을 사용하여 사용자 인증을 참조하세요.

- 자세한 API 내용은 명령 참조CreateUser의 섹션을 참조하세요. AWS CLI

## delete-acl

다음 코드 예시에서는 delete-acl을 사용하는 방법을 보여 줍니다.

AWS CLI

를 삭제하려면 ACL

다음 delete-acl 예제에서는 액세스 제어 목록을 삭제합니다.

```
aws memorydb delete-acl \
    --acl-name "new-acl-1"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 MemoryDB 사용 설명서의 액세스 제어 목록을 사용하여 사용자 인증을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteAcl의 섹션을 참조하세요. AWS CLI

### `delete-cluster`

다음 코드 예시에서는 `delete-cluster`을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터를 삭제하려면

다음 `delete-cluster` 예제에서는 클러스터를 삭제합니다.

```
aws memorydb delete-cluster \
    --cluster-name my-new-cluster
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 MemoryDB 사용 설명서의 클러스터 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteCluster의 섹션을 참조하세요. AWS CLI

### delete-parameter-group

다음 코드 예시에서는 delete-parameter-group을 사용하는 방법을 보여 줍니다.

AWS CLI

파라미터 그룹을 삭제하려면

다음 delete-parameter-group 예제에서는 파라미터 그룹을 삭제합니다.

```
aws memorydb delete-parameter-group \
    --parameter-group-name myRedis6x
```

출력:

```
{
    "ParameterGroup": {
        "Name": "myredis6x",
        "Family": "memorydb_redis6",
```

```
        "Description": "my-parameter-group",
        "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:parametergroup/myredis6x"
    }
}
```

자세한 내용은 MemoryDB 사용 설명서의 파라미터 그룹 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteParameterGroup의 섹션을 참조하세요. AWS CLI

### `delete-snapshot`

다음 코드 예시에서는 `delete-snapshot`을 사용하는 방법을 보여 줍니다.

AWS CLI

스냅샷을 삭제하는 방법

다음 `delete-snapshot` 예제에서는 스냅샷을 삭제합니다.

```
aws memorydb delete-snapshot \
    --snapshot-name my-cluster-snapshot
```

출력:

```
{
    "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
        }
    }
}
```

자세한 내용은 MemoryDB 사용 설명서의 스냅샷 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteSnapshot의 섹션을 참조하세요. AWS CLI

### `delete-subnet-group`

다음 코드 예시에서는 `delete-subnet-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

서브넷 그룹을 삭제하려면

다음 `delete-subnet-group` 예제에서는 서브넷을 삭제합니다.

```
aws memorydb delete-subnet-group \
    --subnet-group-name mysubnetgroup
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 MemoryDB 사용 설명서의 서브넷 그룹 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteSubnetGroup의 섹션을 참조하세요. AWS CLI

### delete-user

다음 코드 예시에서는 delete-user을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 삭제

다음 delete-user 예제에서는 사용자를 삭제합니다.

```
aws memorydb delete-user \
    --user-name my-user
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 MemoryDB 사용 설명서의 액세스 제어 목록을 사용하여 사용자 인증을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteUser의 섹션을 참조하세요. AWS CLI

### describe-acls

다음 코드 예시에서는 describe-acls을 사용하는 방법을 보여 줍니다.

AWS CLI

목록을 반환하려면 ACLs

다음 describe-acls`는 목록을 반환합니다ACLs.

```
aws memorydb describe-acls
```

출력:

```
{
    "ACLs": [
        {
            "Name": "open-access",
            "Status": "active",
            "UserNames": [
                "default"
            ],
            "MinimumEngineVersion": "6.2",
            "Clusters": [],
            "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:acl/open-access"
        },
        {
            "Name": my-acl",
            "Status": "active",
            "UserNames": [],
            "MinimumEngineVersion": "6.2",
            "Clusters": [
                "my-cluster"
            ],
            "ARN": "arn:aws:memorydb:us-east-1:49165xxxxxxx:acl/my-acl"
        }
    ]
}
```

자세한 내용은 MemoryDB 사용 설명서의 액세스 제어 목록을 사용하여 사용자 인증을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeAcls의 섹션을 참조하세요. AWS CLI

### `describe-clusters`

다음 코드 예시에서는 `describe-clusters`을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터 목록을 반환하려면

다음 describe-clusters`는 클러스터 목록을 반환합니다.

```
aws memorydb describe-clusters
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 MemoryDB 사용 설명서의 클러스터 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeClusters의 섹션을 참조하세요. AWS CLI

### describe-engine-versions

다음 코드 예시에서는 describe-engine-versions을 사용하는 방법을 보여 줍니다.

AWS CLI

엔진 버전 목록을 반환하려면

다음 describe-engine-versions`는 엔진 버전 목록을 반환합니다.

```
aws memorydb describe-engine-versions
```

출력:

```
{
    "EngineVersions": [
        {
            "EngineVersion": "6.2",
            "EnginePatchVersion": "6.2.6",
            "ParameterGroupFamily": "memorydb_redis6"
        }
    ]
}
```

자세한 내용은 MemoryDB 사용 설명서의 엔진 버전 및 업그레이드를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeEngineVersions의 섹션을 참조하세요. AWS CLI

### describe-events

다음 코드 예시에서는 describe-events을 사용하는 방법을 보여 줍니다.

AWS CLI

이벤트 목록을 반환하려면

다음 describe-events`는 이벤트 목록을 반환합니다.

```
aws memorydb describe-events
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 MemoryDB 사용 설명서의 이벤트 모니터링을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeEvents의 섹션을 참조하세요. AWS CLI

### describe-parameter-groups

다음 코드 예시에서는 describe-parameter-groups을 사용하는 방법을 보여 줍니다.

AWS CLI

파라미터 그룹 목록을 반환하려면

다음 describe-parameter-groups`는 파라미터 그룹 목록을 반환합니다.

```
aws memorydb describe-parameter-groups
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 MemoryDB 사용 설명서의 파라미터 그룹을 사용하여 엔진 파라미터 구성을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeParameterGroups의 섹션을 참조하세요. AWS CLI

### `describe-parameters`

다음 코드 예시에서는 `describe-parameters`을 사용하는 방법을 보여 줍니다.

AWS CLI

파라미터 목록을 반환하려면

다음 describe-parameters`는 파라미터 목록을 반환합니다.

```
aws memorydb describe-parameters
```

출력:

```
{
    "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": "activerehashing",
            "Value": "yes",
            "Description": "Apply rehashing or not",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "client-output-buffer-limit-normal-hard-limit",
            "Value": "0",
            "Description": "Normal client output buffer hard limit in bytes",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "client-output-buffer-limit-normal-soft-limit",
            "Value": "0",
            "Description": "Normal client output buffer soft limit in bytes",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "client-output-buffer-limit-normal-soft-seconds",
```

```
            "Value": "0",
            "Description": "Normal client output buffer soft limit in seconds",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "client-output-buffer-limit-pubsub-hard-limit",
            "Value": "33554432",
            "Description": "Pubsub client output buffer hard limit in bytes",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "client-output-buffer-limit-pubsub-soft-limit",
            "Value": "8388608",
            "Description": "Pubsub client output buffer soft limit in bytes",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "client-output-buffer-limit-pubsub-soft-seconds",
            "Value": "60",
            "Description": "Pubsub client output buffer soft limit in seconds",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "hash-max-ziplist-entries",
            "Value": "512",
            "Description": "The maximum number of hash entries in order for the
dataset to be compressed",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "hash-max-ziplist-value",
            "Value": "64",
            "Description": "The threshold of biggest hash entries in order for the
dataset to be compressed",
```

```
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "hll-sparse-max-bytes",
            "Value": "3000",
            "Description": "HyperLogLog sparse representation bytes limit",
            "DataType": "integer",
            "AllowedValues": "1-16000",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "lazyfree-lazy-eviction",
            "Value": "no",
            "Description": "Perform an asynchronous delete on evictions",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "lazyfree-lazy-expire",
            "Value": "no",
            "Description": "Perform an asynchronous delete on expired keys",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "lazyfree-lazy-server-del",
            "Value": "no",
            "Description": "Perform an asynchronous delete on key updates",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "lazyfree-lazy-user-del",
            "Value": "no",
            "Description": "Specifies whether the default behavior of DEL command
acts the same as UNLINK",
            "DataType": "string",
            "AllowedValues": "yes,no",
            "MinimumEngineVersion": "6.2.4"
```

```
        },
        {
            "Name": "lfu-decay-time",
            "Value": "1",
            "Description": "The amount of time in minutes to decrement the key
 counter for LFU eviction policyd",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "lfu-log-factor",
            "Value": "10",
            "Description": "The log factor for incrementing key counter for LFU
 eviction policy",
            "DataType": "integer",
            "AllowedValues": "1-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "list-compress-depth",
            "Value": "0",
            "Description": "Number of quicklist ziplist nodes from each side of
 the list to exclude from compression. The head and tail of the list are always
 uncompressed for fast push/pop operations",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "maxmemory-policy",
            "Value": "noeviction",
            "Description": "Max memory policy",
            "DataType": "string",
            "AllowedValues": "volatile-lru,allkeys-lru,volatile-lfu,allkeys-
lfu,volatile-random,allkeys-random,volatile-ttl,noeviction",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "maxmemory-samples",
            "Value": "3",
            "Description": "Max memory samples",
            "DataType": "integer",
            "AllowedValues": "1-",
```

```
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "notify-keyspace-events",
            "Description": "The keyspace events for Redis to notify Pub/Sub clients
about. By default all notifications are disabled",
            "DataType": "string",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "set-max-intset-entries",
            "Value": "512",
            "Description": "The limit in the size of the set in order for the
dataset to be compressed",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "slowlog-log-slower-than",
            "Value": "10000",
            "Description": "The execution time, in microseconds, to exceed in order
for the command to get logged. Note that a negative number disables the slow log,
while a value of zero forces the logging of every command",
            "DataType": "integer",
            "AllowedValues": "-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "slowlog-max-len",
            "Value": "128",
            "Description": "The length of the slow log. There is no limit to this
length. Just be aware that it will consume memory. You can reclaim memory used by
the slow log with SLOWLOG RESET.",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "stream-node-max-bytes",
            "Value": "4096",
            "Description": "The maximum size of a single node in a stream in bytes",
            "DataType": "integer",
            "AllowedValues": "0-",
```

```
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "stream-node-max-entries",
            "Value": "100",
            "Description": "The maximum number of items a single node in a stream
can contain",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "tcp-keepalive",
            "Value": "300",
            "Description": "If non-zero, send ACKs every given number of seconds",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "timeout",
            "Value": "0",
            "Description": "Close connection if client is idle for a given number of
seconds, or never if 0",
            "DataType": "integer",
            "AllowedValues": "0,20-",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "tracking-table-max-keys",
            "Value": "1000000",
            "Description": "The maximum number of keys allowed for the tracking
table for client side caching",
            "DataType": "integer",
            "AllowedValues": "1-100000000",
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "zset-max-ziplist-entries",
            "Value": "128",
            "Description": "The maximum number of sorted set entries in order for
the dataset to be compressed",
            "DataType": "integer",
            "AllowedValues": "0-",
```

```
            "MinimumEngineVersion": "6.2.4"
        },
        {
            "Name": "zset-max-ziplist-value",
            "Value": "64",
            "Description": "The threshold of biggest sorted set entries in order for
 the dataset to be compressed",
            "DataType": "integer",
            "AllowedValues": "0-",
            "MinimumEngineVersion": "6.2.4"
        }
    ]
}
```

자세한 내용은 MemoryDB 사용 설명서의 파라미터 그룹을 사용하여 엔진 파라미터 구성을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeParameters의 섹션을 참조하세요. AWS CLI

### `describe-snapshots`

다음 코드 예시에서는 `describe-snapshots`을 사용하는 방법을 보여 줍니다.

AWS CLI

스냅샷 목록을 반환하려면

다음 describe-snapshots`는 스냅샷 목록을 반환합니다.

```
aws memorydb describe-snapshots
```

출력:

```
{
"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
        }
    }
}
```

자세한 내용은 MemoryDB 사용 설명서의 스냅샷 및 복원을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeSnapshots의 섹션을 참조하세요. AWS CLI

### `describe-subnet-groups`

다음 코드 예시에서는 `describe-subnet-groups`을 사용하는 방법을 보여 줍니다.

AWS CLI

서브넷 그룹 목록을 반환하려면

다음 describe-subnet-groups`는 서브넷 그룹 목록을 반환합니다.

```
aws memorydb describe-subnet-groups
```

출력

```
{
    "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"
        }
    ]
}
```

자세한 내용은 MemoryDB 사용 설명서의 서브넷 및 서브넷 그룹을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeSubnetGroups의 섹션을 참조하세요. AWS CLI

### `describe-users`

다음 코드 예시에서는 `describe-users`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 목록을 반환하려면

다음 describe-users`는 사용자 목록을 반환합니다.

```
aws memorydb describe-users
```

출력

```
{
    "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"
        }
    ]
}
```

자세한 내용은 MemoryDB 사용 설명서의 액세스 제어 목록을 사용하여 사용자 인증을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeUsers의 섹션을 참조하세요. AWS CLI

### `failover-shard`

다음 코드 예시에서는 `failover-shard`을 사용하는 방법을 보여 줍니다.

AWS CLI

샤드를 장애 조치하려면

다음 장애 조치 샤드는 샤드를 통해 실패합니다.

```
aws memorydb failover-shard \
    --cluster-name my-cluster --shard-name 0001
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 MemoryDB 사용 설명서의 MultiAZ를 사용하여 가동 중지 시간 최소화를 참조하세요.

- 자세한 API 내용은 명령 참조FailoverShard의 섹션을 참조하세요. AWS CLI

### `list-allowed-node-type-updates`

다음 코드 예시에서는 `list-allowed-node-type-updates`을 사용하는 방법을 보여 줍니다.

AWS CLI

허용된 노드 유형 업데이트 목록을 반환하려면

다음 list-allowed-node-type업데이트는 사용 가능한 노드 유형 업데이트 목록을 반환합니다.

```
aws memorydb list-allowed-node-type-updates
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 MemoryDB 사용 설명서의 크기 조정을 참조하세요.

- 자세한 API 내용은 명령 참조ListAllowedNodeTypeUpdates의 섹션을 참조하세요. AWS CLI

### `list-tags`

다음 코드 예시에서는 `list-tags`을 사용하는 방법을 보여 줍니다.

AWS CLI

태그 목록을 반환하려면

다음 list-tags는 태그 목록을 반환합니다.

```
aws memorydb list-tags \
    --resource-arn arn:aws:memorydb:us-east-1:491658xxxxxx:cluster/my-cluster
```

출력:

```
{
    "TagList": [
        {
            "Key": "mytag",
            "Value": "myvalue"
        }
    ]
}
```

자세한 내용은 MemoryDB 사용 설명서의 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조ListTags의 섹션을 참조하세요. AWS CLI

### reset-parameter-group

다음 코드 예시에서는 reset-parameter-group을 사용하는 방법을 보여 줍니다.

AWS CLI

파라미터 그룹을 재설정하려면

다음 reset-parameter-group`는 파라미터 그룹을 재설정합니다.

```
aws memorydb reset-parameter-group \
    --parameter-group-name my-parameter-group \
    --all-parameters
```

출력:

```
{
```

```
    "ParameterGroup": {
        "Name": "my-parameter-group",
        "Family": "memorydb_redis6",
        "Description": "my parameter group",
        "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:parametergroup/my-parameter-
group"
    }
}
```

자세한 내용은 MemoryDB 사용 설명서의 파라미터 그룹을 사용하여 엔진 파라미터 구성을 참조하세요.

- 자세한 API 내용은 명령 참조ResetParameterGroup의 섹션을 참조하세요. AWS CLI

### `tag-resource`

다음 코드 예시에서는 `tag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 태그를 지정하려면

다음 tag-resource`는 리소스에 태그를 추가합니다.

```
aws memorydb tag-resource \
    --resource-arn arn:aws:memorydb:us-east-1:491658xxxxxx:cluster/my-cluster \
    --tags Key="mykey",Value="myvalue"
```

출력:

```
{
    "TagList": [
        {
            "Key": "mytag",
            "Value": "myvalue"
        },
        {
            "Key": "mykey",
            "Value": "myvalue"
        }
    ]
}
```

자세한 내용은 MemoryDB 사용 설명서의 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### `untag-resource`

다음 코드 예시에서는 `untag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

를 업데이트하려면 ACL

다음 update-acl`은 사용자를 ACL 추가하여 를 업데이트합니다.

```
aws memorydb untag-resource \
    --resource-arn arn:aws:memorydb:us-east-1:491658xxxxx:cluster/my-cluster \
    --tag-keys mykey
```

출력:

```
{
    "TagList": [
        {
            "Key": "mytag",
            "Value": "myvalue"
        }
    ]
}
```

자세한 내용은 MemoryDB 사용 설명서의 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### `update-cluster`

다음 코드 예시에서는 `update-cluster`을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터를 업데이트하려면

다음 update-cluster``는 클러스터의 파라미터 그룹을 로 업데이트합니다 my-parameter-group.

```
aws memorydb update-cluster \
    --cluster-name my-cluster \
    --parameter-group-name my-parameter-group
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 MemoryDB 사용 설명서의 클러스터 수정을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateCluster의 섹션을 참조하세요. AWS CLI

### update-parameter-group

다음 코드 예시에서는 update-parameter-group을 사용하는 방법을 보여 줍니다.

AWS CLI

파라미터 그룹을 업데이트하려면

다음 update-parameter-group``는 파라미터 그룹을 업데이트합니다.

```
aws memorydb update-parameter-group \
    --parameter-group-name my-parameter-group \
    --parameter-name-values "ParameterName=activedefrag, ParameterValue=no"
```

출력:

```
{
    "ParameterGroup": {
        "Name": "my-parameter-group",
        "Family": "memorydb_redis6",
        "Description": "my parameter group",
        "ARN": "arn:aws:memorydb:us-east-1:49165xxxxxx:parametergroup/my-parameter-
group"
    }
}
```

자세한 내용은 MemoryDB 사용 설명서의 파라미터 그룹 수정을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateParameterGroup의 섹션을 참조하세요. AWS CLI

### update-subnet-group

다음 코드 예시에서는 update-subnet-group을 사용하는 방법을 보여 줍니다.

AWS CLI

서브넷 그룹을 업데이트하려면

다음 update-subnet-group`는 서브넷 그룹의 서브넷 ID를 업데이트합니다.

```
aws memorydb update-subnet-group \
    --subnet-group-name my-sg \
    --subnet-ids subnet-01f29d458f3xxxxx
```

출력:

```
{
    "SubnetGroup": {
        "Name": "my-sg-1",
        "Description": "my-sg",
        "VpcId": "vpc-09d2cfc01xxxxxxx",
        "Subnets": [
            {
                "Identifier": "subnet-01f29d458fxxxxxx",
                "AvailabilityZone": {
                    "Name": "us-east-1a"
                }
            }
        ],
        "ARN": "arn:aws:memorydb:us-east-1:491658xxxxxx:subnetgroup/my-sg"
    }
}
```

자세한 내용은 MemoryDB 사용 설명서의 서브넷 및 서브넷 그룹을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateSubnetGroup의 섹션을 참조하세요. AWS CLI

### update-user

다음 코드 예시에서는 update-user을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자를 업데이트하려면

다음은 사용자의 액세스 문자열을 update-user 수정합니다.

```
aws memorydb update-user \
    --user-name my-user \
    --access-string "off ~objects:* ~items:* ~public:* resetchannels -@all"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 MemoryDB 사용 설명서의 액세스 제어 목록을 사용하여 사용자 인증을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateUser의 섹션을 참조하세요. AWS CLI

## 를 사용한 Amazon MSK 예제 AWS CLI

다음 코드 예제에서는 Amazon 와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다MSK.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `create-cluster`

다음 코드 예시에서는 `create-cluster`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon MSK 클러스터를 생성하려면

다음 create-cluster 예제에서는 3개의 브로커 노드MessagingCluster로 라는 MSK 클러스터를 생성합니다. 이름이 인 JSON 파일은 Amazon이 브로커 노드를 배포할 3개의 서브넷MSK을 brokernodegroupinfo.json 지정합니다. 이 예제에서는 모니터링 수준을 지정하지 않으므로 클러스터가 DEFAULT 수준을 가져옵니다.

```
aws kafka create-cluster \
    --cluster-name "MessagingCluster" \
    --broker-node-group-info file://brokernodegroupinfo.json \
    --kafka-version "2.2.1" \
    --number-of-broker-nodes 3
```

brokernodegroupinfo.json의 콘텐츠:

```
{
    "InstanceType": "kafka.m5.xlarge",
    "BrokerAZDistribution": "DEFAULT",
    "ClientSubnets": [
        "subnet-0123456789111abcd",
        "subnet-0123456789222abcd",
        "subnet-0123456789333abcd"
    ]
}
```

출력:

```
{
    "ClusterArn": "arn:aws:kafka:us-west-2:123456789012:cluster/MessagingCluster/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE-2",
    "ClusterName": "MessagingCluster",
    "State": "CREATING"
}
```

자세한 내용은 Amazon Managed Streaming for Apache Kafka의 Amazon MSK 클러스터 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateCluster의 섹션을 참조하세요. AWS CLI

### create-configuration

다음 코드 예시에서는 create-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 Amazon MSK 구성을 생성하려면

다음 create-configuration 예제에서는 입력 파일에 지정된 서버 속성을 사용하여 사용자 지정 MSK 구성을 생성합니다.

```
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
```

configuration.txt의 콘텐츠:

```
auto.create.topics.enable = true
zookeeper.connection.timeout.ms = 2000
log.roll.ms = 604800000
```

이 명령은 출력을 생성하지 않습니다. 출력:

```
{
    "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"
}
```

자세한 내용은 Amazon Managed Streaming for Apache Kafka 개발자 안내서의 Amazon MSK 구성 작업을 참조하세요.

- 자세한 API 내용은 명령 참조CreateConfiguration의 섹션을 참조하세요. AWS CLI

### describe-cluster

다음 코드 예시에서는 describe-cluster을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터를 설명하려면

다음 describe-cluster 예제에서는 Amazon MSK 클러스터를 설명합니다.

```
aws kafka describe-cluster \
    --cluster-arn arn:aws:kafka:us-east-1:123456789012:cluster/demo-
cluster-1/6357e0b2-0e6a-4b86-a0b4-70df934c2e31-5
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Managed Streaming for Apache Kafka 개발자 안내서의 Amazon MSK 클러스터 나열을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeCluster의 섹션을 참조하세요. AWS CLI

### `get-bootstrap-brokers`

다음 코드 예시에서는 `get-bootstrap-brokers`을 사용하는 방법을 보여 줍니다.

AWS CLI

부트스트랩 브로커를 가져오려면

다음 get-bootstrap-brokers 예제에서는 Amazon MSK 클러스터에 대한 부트스트랩 브로커 정보를 검색합니다.

```
aws kafka get-bootstrap-brokers \
    --cluster-arn arn:aws:kafka:us-east-1:123456789012:cluster/demo-cluster-1/6357e0b2-0e6a-4b86-a0b4-70df934c2e31-5
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon Managed Streaming for Apache Kafka 개발자 안내서의 부트스트랩 브로커 가져오기를 참조하세요.

- 자세한 API 내용은 명령 참조GetBootstrapBrokers의 섹션을 참조하세요. AWS CLI

### list-clusters

다음 코드 예시에서는 list-clusters을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 클러스터를 나열하려면

다음 list-clusters 예제에서는 AWS 계정의 Amazon MSK 클러스터를 나열합니다.

```
aws kafka list-clusters
```

출력:

```
{
    "ClusterInfoList": [
        {
            "BrokerNodeGroupInfo": {
                "BrokerAZDistribution": "DEFAULT",
```

```
                "ClientSubnets": [
                    "subnet-cbfff283",
                    "subnet-6746046b"
                ],
                "InstanceType": "kafka.m5.large",
                "SecurityGroups": [
                    "sg-f839b688"
                ],
                "StorageInfo": {
                    "EbsStorageInfo": {
                        "VolumeSize": 100
                    }
                }
            },
            "ClusterArn": "arn:aws:kafka:us-east-1:123456789012:cluster/demo-
cluster-1/6357e0b2-0e6a-4b86-a0b4-70df934c2e31-5",
            "ClusterName": "demo-cluster-1",
            "CreationTime": "2020-07-09T02:31:36.223000+00:00",
            "CurrentBrokerSoftwareInfo": {
                "KafkaVersion": "2.2.1"
            },
            "CurrentVersion": "K3AEGXETSR30VB",
            "EncryptionInfo": {
                "EncryptionAtRest": {
                    "DataVolumeKMSKeyId": "arn:aws:kms:us-east-1:123456789012:key/
a7ca56d5-0768-4b64-a670-339a9fbef81c"
                },
                "EncryptionInTransit": {
                    "ClientBroker": "TLS_PLAINTEXT",
                    "InCluster": true
                }
            },
            "EnhancedMonitoring": "DEFAULT",
            "OpenMonitoring": {
                "Prometheus": {
                    "JmxExporter": {
                        "EnabledInBroker": false
                    },
                    "NodeExporter": {
                        "EnabledInBroker": false
                    }
                }
            },
            "NumberOfBrokerNodes": 2,
```

```
            "State": "ACTIVE",
            "Tags": {},
            "ZookeeperConnectString": "z-2.demo-cluster-1.xuy0sb.c5.kafka.us-
east-1.amazonaws.com:2181,z-1.demo-cluster-1.xuy0sb.c5.kafka.us-
east-1.amazonaws.com:2181,z-3.demo-cluster-1.xuy0sb.c5.kafka.us-
east-1.amazonaws.com:2181"
        }
    ]
}
```

자세한 내용은 Amazon Managed Streaming for Apache Kafka 개발자 안내서의 Amazon MSK 클러스터 나열을 참조하세요.

- 자세한 API 내용은 명령 참조ListClusters의 섹션을 참조하세요. AWS CLI

### update-broker-storage

다음 코드 예시에서는 update-broker-storage을 사용하는 방법을 보여 줍니다.

AWS CLI

브로커의 EBS 스토리지를 업데이트하려면

다음 update-broker-storage 예제에서는 클러스터의 모든 브로커에 대한 EBS 스토리지 양을 업데이트합니다. Amazon은 각 브로커의 목표 스토리지 양을 예제에 지정된 양으로 MSK 설정합니다. 클러스터를 설명하거나 모든 클러스터를 나열하여 클러스터의 현재 버전을 가져올 수 있습니다.

```
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"
```

출력은 이 update-broker-storage 작업에 ARN 대해 를 반환합니다. 이 작업이 완료되었는지 확인하려면 이 describe-cluster-operation 명령과 함께 를 입력ARN으로 사용합니다.

```
{
    "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"
}
```

자세한 내용은 Amazon Managed Streaming for Apache Kafka 개발자 안내서의 브로커용 EBS 스토리지 업데이트를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateBrokerStorage의 섹션을 참조하세요. AWS CLI

### update-cluster-configuration

다음 코드 예시에서는 update-cluster-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon MSK 클러스터의 구성을 업데이트하려면

다음 update-cluster-configuration 예제에서는 지정된 기존 MSK 클러스터의 구성을 업데이트합니다. 사용자 지정 MSK 구성을 사용합니다.

```
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"
```

configuration-info.json의 콘텐츠:

```
{
    "Arn": "arn:aws:kafka:us-west-2:123456789012:configuration/CustomConfiguration/
a1b2c3d4-5678-90ab-cdef-11111EXAMPLE-2",
    "Revision": 1
}
```

출력은 이 update-cluster-configuration 작업에 ARN 대해 를 반환합니다. 이 작업이 완료되었는지 확인하려면 이 describe-cluster-operation 명령과 함께 를 입력ARN으로 사용합니다.

```
{
    "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"
}
```

자세한 내용은 Amazon Managed Streaming for Apache Kafka 개발자 안내서의 Amazon MSK 클러스터 구성 업데이트를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateClusterConfiguration의 섹션을 참조하세요. AWS CLI

## 를 사용한 Network Manager 예제 AWS CLI

다음 코드 예제에서는 Network Manager AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `associate-customer-gateway`

다음 코드 예시에서는 `associate-customer-gateway`을 사용하는 방법을 보여 줍니다.

AWS CLI

고객 게이트웨이를 연결하려면

다음 `associate-customer-gateway` 예제는 지정된 글로벌 네트워크의 고객 게이트웨이cgw-11223344556677889를 디바이스 와 연결합니다device-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
```

출력:

```
{
    "CustomerGatewayAssociation": {
        "CustomerGatewayArn": "arn:aws:ec2:us-west-2:123456789012:customer-gateway/
cgw-11223344556677889",
        "GlobalNetworkId": "global-network-01231231231231231",
        "DeviceId": "device-07f6fd08867abc123",
        "State": "PENDING"
    }
}
```

자세한 내용은 Transit Gateway Network Manager 안내서의 Customer Gateway Associations를 참조하세요.

- 자세한 API 내용은 명령 참조AssociateCustomerGateway의 섹션을 참조하세요. AWS CLI

### associate-link

다음 코드 예시에서는 associate-link을 사용하는 방법을 보여 줍니다.

AWS CLI

링크를 연결하려면

다음 associate-link 예제는 링크를 디바이스 link-11112222aaaabbbb1와 연결합니다device-07f6fd08867abc123. 링크와 디바이스는 지정된 글로벌 네트워크에 있습니다.

```
aws networkmanager associate-link \
    --global-network-id global-network-01231231231231231 \
    --device-id device-07f6fd08867abc123 \
    --link-id link-11112222aaaabbbb1 \
    --region us-west-2
```

출력:

```
{
    "LinkAssociation": {
```

```
        "GlobalNetworkId": "global-network-01231231231231231",
        "DeviceId": "device-07f6fd08867abc123",
        "LinkId": "link-11112222aaaabbbb1",
        "LinkAssociationState": "PENDING"
    }
}
```

자세한 내용은 Transit Gateway Network Manager 가이드의 디바이스 및 링크 연결을 참조하세요.

- 자세한 API 내용은 명령 참조AssociateLink의 섹션을 참조하세요. AWS CLI

### `create-core-network`

다음 코드 예시에서는 `create-core-network`을 사용하는 방법을 보여 줍니다.

AWS CLI

코어 네트워크를 생성하려면

다음 `create-core-network` 예제에서는 AWS Cloud WAN 글로벌 네트워크 내에서 선택적 설명과 태그를 사용하여 코어 네트워크를 생성합니다.

```
aws networkmanager create-core-network \
    --global-network-id global-network-cdef-EXAMPLE22222 \
    --description "Main headquarters location" \
    --tags Key=Name,Value="New York City office"
```

출력:

```
{
    "CoreNetwork": {
        "GlobalNetworkId": "global-network-cdef-EXAMPLE22222",
        "CoreNetworkId": "core-network-cdef-EXAMPLE33333",
        "CoreNetworkArn": "arn:aws:networkmanager::987654321012:core-network/core-
network-cdef-EXAMPLE33333",
        "Description": "Main headquarters location",
        "CreatedAt": "2022-01-10T19:53:59+00:00",
        "State": "AVAILABLE",
        "Tags": [
            {
                "Key": "Name",
                "Value": "New York City office"
```

```
            }
        ]
    }
}
```

자세한 내용은 AWS 클라우드 WAN 사용 설명서의 글로벌 및 코어 네트워크를 참조하세요.

- 자세한 API 내용은 명령 참조CreateCoreNetwork의 섹션을 참조하세요. AWS CLI

### `create-device`

다음 코드 예시에서는 `create-device`을 사용하는 방법을 보여 줍니다.

AWS CLI

디바이스를 생성하려면

다음 `create-device` 예제에서는 지정된 글로벌 네트워크에 디바이스를 생성합니다. 디바이스 세부 정보에는 설명, 유형, 공급업체, 모델 및 일련 번호가 포함됩니다.

```
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
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Transit Gateway Network Manager 가이드의 디바이스 작업을 참조하세요.

- 자세한 API 내용은 명령 참조CreateDevice의 섹션을 참조하세요. AWS CLI

### `create-global-network`

다음 코드 예시에서는 `create-global-network`을 사용하는 방법을 보여 줍니다.

AWS CLI

글로벌 네트워크를 생성하려면

다음 `create-global-network` 예제에서는 새 글로벌 네트워크를 생성합니다. 생성 시 초기 상태는 입니다PENDING.

```
aws networkmanager create-global-network
```

출력:

```
{
    "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"
    }
}
```

- 자세한 API 내용은 명령 참조CreateGlobalNetwork의 섹션을 참조하세요. AWS CLI

### `create-link`

다음 코드 예시에서는 `create-link`을 사용하는 방법을 보여 줍니다.

AWS CLI

링크를 생성하려면

다음 create-link 예제에서는 지정된 글로벌 네트워크에 링크를 생성합니다. 링크에는 링크 유형, 대역폭 및 공급자에 대한 설명과 세부 정보가 포함되어 있습니다. 사이트 ID는 링크가 연결된 사이트를 나타냅니다.

```
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
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Transit Gateway Network Manager 가이드의 링크 작업을 참조하세요.

- 자세한 API 내용은 명령 참조CreateLink의 섹션을 참조하세요. AWS CLI

### create-site

다음 코드 예시에서는 create-site을 사용하는 방법을 보여 줍니다.

AWS CLI

사이트를 생성하려면

다음 create-site 예제에서는 지정된 글로벌 네트워크에 사이트를 생성합니다. 사이트 세부 정보에는 설명과 위치 정보가 포함됩니다.

```
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
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Transit Gateway Network Manager 가이드의 사이트 작업을 참조하세요.

- 자세한 API 내용은 명령 참조CreateSite의 섹션을 참조하세요. AWS CLI

### create-vpc-attachment

다음 코드 예시에서는 create-vpc-attachment을 사용하는 방법을 보여 줍니다.

AWS CLI

VPC 첨부 파일을 생성하려면

다음 create-vpc-attachment 예제에서는 코어 네트워크에서 IPv6 지원이 포함된 VPC 연결을 생성합니다.

```
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
```

출력:

```
{
    "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
        }
    }
}
```

자세한 내용은 클라우드 WAN 사용 설명서의 첨부 파일 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateVpcAttachment의 섹션을 참조하세요. AWS CLI

### delete-attachment

다음 코드 예시에서는 delete-attachment을 사용하는 방법을 보여 줍니다.

AWS CLI

첨부 파일을 삭제하려면

다음 delete-attachment 예제에서는 Connect 연결을 삭제합니다.

```
aws networkmanager delete-attachment \
    --attachment-id attachment-01feddaeae26ab68c
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 클라우드 WAN 사용 설명서의 첨부 파일 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteAttachment의 섹션을 참조하세요. AWS CLI

### delete-bucket-analytics-configuration

다음 코드 예시에서는 delete-bucket-analytics-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

버킷의 분석 구성을 삭제하려면

다음 delete-bucket-analytics-configuration 예시는 지정된 버킷 및 ID에 대한 분석 구성을 제거합니다.

```
aws s3api delete-bucket-analytics-configuration \
    --bucket my-bucket \
    --id 1
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeleteBucketAnalyticsConfiguration의 섹션을 참조하세요. AWS CLI

### delete-bucket-metrics-configuration

다음 코드 예시에서는 delete-bucket-metrics-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

버킷의 지표 구성을 삭제하려면

다음 delete-bucket-metrics-configuration 예시는 지정된 버킷 및 ID에 대한 지표 구성을 제거합니다.

```
aws s3api delete-bucket-metrics-configuration \
    --bucket my-bucket \
    --id 123
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeleteBucketMetricsConfiguration의 섹션을 참조하세요. AWS CLI

### delete-core-network

다음 코드 예시에서는 delete-core-network을 사용하는 방법을 보여 줍니다.

AWS CLI

코어 네트워크를 삭제하려면

다음 delete-core-network 예제에서는 Cloud WAN 글로벌 네트워크에서 코어 네트워크를 삭제합니다.

```
aws networkmanager delete-core-network \
    --core-network-id core-network-0fab62fe438d94db6
```

출력:

```
{
    "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": []
            }
        ]
    }
}
```

자세한 내용은 클라우드 WAN 사용 설명서의 코어 네트워크를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteCoreNetwork의 섹션을 참조하세요. AWS CLI

### `delete-device`

다음 코드 예시에서는 `delete-device`을 사용하는 방법을 보여 줍니다.

AWS CLI

디바이스를 삭제하려면

다음 delete-device 예제에서는 지정된 글로벌 네트워크에서 지정된 디바이스를 삭제합니다.

```
aws networkmanager delete-device \
    --global-network-id global-network-01231231231231231 \
    --device-id device-07f6fd08867abc123 \
    --region us-west-2
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Transit Gateway Network Manager 가이드의 디바이스 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteDevice의 섹션을 참조하세요. AWS CLI

### delete-global-network

다음 코드 예시에서는 delete-global-network을 사용하는 방법을 보여 줍니다.

AWS CLI

글로벌 네트워크를 삭제하려면

다음 delete-global-network 예제에서는 글로벌 네트워크를 삭제합니다.

```
aws networkmanager delete-global-network \
    --global-network-id global-network-052bedddccb193b6b
```

출력:

```
{
    "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"
    }
}
```

- 자세한 API 내용은 명령 참조DeleteGlobalNetwork의 섹션을 참조하세요. AWS CLI

### delete-link

다음 코드 예시에서는 delete-link을 사용하는 방법을 보여 줍니다.

AWS CLI

링크를 삭제하려면

다음 delete-link 예제에서는 지정된 글로벌 네트워크에서 지정된 링크를 삭제합니다.

```
aws networkmanager delete-link \
    --global-network-id global-network-01231231231231231  \
    --link-id link-11112222aaaabbbb1 \
    --region us-west-2
```

출력:

```
{
    "Link": {
        "LinkId": "link-11112222aaaabbbb1",
        "LinkArn": "arn:aws:networkmanager::123456789012:link/global-
network-01231231231231231/link-11112222aaaabbbb1",
        "GlobalNetworkId": "global-network-01231231231231231",
        "SiteId": "site-444555aaabbb11223",
        "Description": "VPN Link",
        "Type": "broadband",
        "Bandwidth": {
            "UploadSpeed": 20,
```

```
            "DownloadSpeed": 20
        },
        "Provider": "AnyCompany",
        "CreatedAt": 1575555811.0,
        "State": "DELETING"
    }
}
```

자세한 내용은 Transit Gateway Network Manager 가이드의 링크 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteLink의 섹션을 참조하세요. AWS CLI

### delete-public-access-block

다음 코드 예시에서는 delete-public-access-block을 사용하는 방법을 보여 줍니다.

AWS CLI

버킷의 퍼블릭 액세스 차단 구성을 삭제하려면

다음 delete-public-access-block 예시는 지정된 버킷에서 퍼블릭 액세스 차단 구성을 제거합니다.

```
aws s3api delete-public-access-block \
    --bucket my-bucket
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeletePublicAccessBlock의 섹션을 참조하세요. AWS CLI

### delete-site

다음 코드 예시에서는 delete-site을 사용하는 방법을 보여 줍니다.

AWS CLI

사이트를 삭제하려면

다음 delete-site 예제에서는 지정된 글로벌 네트워크에서 지정된 사이트(site-444555aaabbb11223)를 삭제합니다.

```
aws networkmanager delete-site \
```

```
    --global-network-id global-network-01231231231231231  \
    --site-id site-444555aaabbb11223 \
    --region us-west-2
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Transit Gateway Network Manager 가이드의 사이트 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteSite의 섹션을 참조하세요. AWS CLI

### deregister-transit-gateway

다음 코드 예시에서는 deregister-transit-gateway을 사용하는 방법을 보여 줍니다.

AWS CLI

글로벌 네트워크에서 전송 게이트웨이 등록을 취소하려면

다음 deregister-transit-gateway 예제에서는 지정된 전역 네트워크에서 지정된 전송 게이트웨이의 등록을 취소합니다.

```
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
```

출력:

```
{
    "TransitGatewayRegistration": {
        "GlobalNetworkId": "global-network-01231231231231231",
        "TransitGatewayArn": "arn:aws:ec2:us-west-2:123456789012:transit-gateway/
tgw-123abc05e04123abc",
        "State": {
            "Code": "DELETING"
        }
    }
}
```

자세한 내용은 Transit Gateway Network Manager 안내서의 Transit Gateway 등록을 참조하세요.

- 자세한 API 내용은 명령 참조DeregisterTransitGateway의 섹션을 참조하세요. AWS CLI

### describe-global-networks

다음 코드 예시에서는 describe-global-networks을 사용하는 방법을 보여 줍니다.

AWS CLI

글로벌 네트워크를 설명하려면

다음 describe-global-networks 예제에서는 계정의 모든 글로벌 네트워크를 설명합니다.

```
aws networkmanager describe-global-networks \
    --region us-west-2
```

출력:

```
{
    "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"
        }
    ]
```

```
}
```

- 자세한 API 내용은 명령 참조DescribeGlobalNetworks의 섹션을 참조하세요. AWS CLI

### disassociate-customer-gateway

다음 코드 예시에서는 disassociate-customer-gateway을 사용하는 방법을 보여 줍니다.

AWS CLI

고객 게이트웨이 연결을 해제하려면

다음 disassociate-customer-gateway 예제에서는 지정된 고객 게이트웨이(cgw-11223344556677889)를 지정된 글로벌 네트워크에서 연결 해제합니다.

```
aws networkmanager disassociate-customer-gateway \
    --global-network-id global-network-01231231231231231 \
    --customer-gateway-arn arn:aws:ec2:us-west-2:123456789012:customer-gateway/
cgw-11223344556677889 \
    --region us-west-2
```

출력:

```
{
    "CustomerGatewayAssociation": {
        "CustomerGatewayArn": "arn:aws:ec2:us-west-2:123456789012:customer-gateway/
cgw-11223344556677889",
        "GlobalNetworkId": "global-network-01231231231231231",
        "DeviceId": "device-07f6fd08867abc123",
        "State": "DELETING"
    }
}
```

자세한 내용은 Transit Gateway Network Manager 안내서의 Customer Gateway Associations를 참조하세요.

- 자세한 API 내용은 명령 참조DisassociateCustomerGateway의 섹션을 참조하세요. AWS CLI

### disassociate-link

다음 코드 예시에서는 disassociate-link을 사용하는 방법을 보여 줍니다.

AWS CLI

링크 연결을 해제하려면

다음 disassociate-link 예제에서는 지정된 글로벌 네트워크의 디바이스에서 device-07f6fd08867abc123 지정된 링크를 연결 해제합니다.

```
aws networkmanager disassociate-link \
    --global-network-id global-network-01231231231231231 \
    --device-id device-07f6fd08867abc123 \
    --link-id link-11112222aaaabbbb1 \
    --region us-west-2
```

출력:

```
{
    "LinkAssociation": {
        "GlobalNetworkId": "global-network-01231231231231231",
        "DeviceId": "device-07f6fd08867abc123",
        "LinkId": "link-11112222aaaabbbb1",
        "LinkAssociationState": "DELETING"
    }
}
```

자세한 내용은 Transit Gateway Network Manager 가이드의 디바이스 및 링크 연결을 참조하세요.

- 자세한 API 내용은 명령 참조DisassociateLink의 섹션을 참조하세요. AWS CLI

## get-bucket-analytics-configuration

다음 코드 예시에서는 get-bucket-analytics-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

특정 ID를 가진 버킷의 분석 구성을 검색하려면

다음 get-bucket-analytics-configuration 예시는 지정된 버킷 및 ID에 대한 분석 구성을 표시합니다.

```
aws s3api get-bucket-analytics-configuration \
    --bucket my-bucket \
```

```
    --id 1
```

출력:

```
{
    "AnalyticsConfiguration": {
        "StorageClassAnalysis": {},
        "Id": "1"
    }
}
```

- 자세한 API 내용은 명령 참조GetBucketAnalyticsConfiguration의 섹션을 참조하세요. AWS CLI

### get-bucket-metrics-configuration

다음 코드 예시에서는 get-bucket-metrics-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

특정 ID를 가진 버킷의 지표 구성을 검색하려면

다음 get-bucket-metrics-configuration 예시는 지정된 버킷 및 ID에 대한 지표 구성을 표시합니다.

```
aws s3api get-bucket-metrics-configuration \
    --bucket my-bucket \
    --id 123
```

출력:

```
{
    "MetricsConfiguration": {
        "Filter": {
            "Prefix": "logs"
        },
        "Id": "123"
    }
}
```

- 자세한 API 내용은 명령 참조GetBucketMetricsConfiguration의 섹션을 참조하세요. AWS CLI

### get-customer-gateway-associations

다음 코드 예시에서는 get-customer-gateway-associations을 사용하는 방법을 보여 줍니다.

AWS CLI

고객 게이트웨이 연결을 가져오려면

다음 get-customer-gateway-associations 예제에서는 지정된 글로벌 네트워크에 대한 고객 게이트웨이 연결을 가져옵니다.

```
aws networkmanager get-customer-gateway-associations \
    --global-network-id global-network-01231231231231231 \
    --region us-west-2
```

출력:

```
{
    "CustomerGatewayAssociations": [
        {
            "CustomerGatewayArn": "arn:aws:ec2:us-west-2:123456789012:customer-
gateway/cgw-11223344556677889",
            "GlobalNetworkId": "global-network-01231231231231231",
            "DeviceId": "device-07f6fd08867abc123",
            "State": "AVAILABLE"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetCustomerGatewayAssociations의 섹션을 참조하세요. AWS CLI

### get-devices

다음 코드 예시에서는 get-devices을 사용하는 방법을 보여 줍니다.

AWS CLI

디바이스를 가져오려면

다음 get-devices 예제에서는 지정된 글로벌 네트워크의 디바이스를 가져옵니다.

```
aws networkmanager get-devices \
```

```
    --global-network-id global-network-01231231231231231 \
    --region us-west-2
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetDevices의 섹션을 참조하세요. AWS CLI

### get-link-associations

다음 코드 예시에서는 get-link-associations을 사용하는 방법을 보여 줍니다.

AWS CLI

링크 연결을 가져오려면

다음 get-link-associations 예제에서는 지정된 글로벌 네트워크의 링크 연결을 가져옵니다.

```
aws networkmanager get-link-associations \
    --global-network-id global-network-01231231231231231 \
    --region us-west-2
```

출력:

```
{
```

```
    "LinkAssociations": [
        {
            "GlobalNetworkId": "global-network-01231231231231231",
            "DeviceId": "device-07f6fd08867abc123",
            "LinkId": "link-11112222aaaabbbb1",
            "LinkAssociationState": "AVAILABLE"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetLinkAssociations의 섹션을 참조하세요. AWS CLI

### get-links

다음 코드 예시에서는 get-links을 사용하는 방법을 보여 줍니다.

AWS CLI

링크를 가져오려면

다음 get-links 예제에서는 지정된 글로벌 네트워크의 링크를 가져옵니다.

```
aws networkmanager get-links \
    --global-network-id global-network-01231231231231231 \
    --region us-west-2
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetLinks의 섹션을 참조하세요. AWS CLI

### get-object-retention

다음 코드 예시에서는 get-object-retention을 사용하는 방법을 보여 줍니다.

AWS CLI

객체의 객체 보존 구성을 검색하는 방법

다음 get-object-retention 예시에서는 지정된 객체에 대한 보존 구성을 검색합니다.

```
aws s3api get-object-retention \
    --bucket my-bucket-with-object-lock \
    --key doc1.rtf
```

출력:

```
{
    "Retention": {
        "Mode": "GOVERNANCE",
        "RetainUntilDate": "2025-01-01T00:00:00.000Z"
    }
}
```

- 자세한 API 내용은 명령 참조GetObjectRetention의 섹션을 참조하세요. AWS CLI

### get-public-access-block

다음 코드 예시에서는 get-public-access-block을 사용하는 방법을 보여 줍니다.

AWS CLI

버킷의 퍼블릭 액세스 차단 구성을 설정하거나 수정하려면

다음 get-public-access-block 예시는 지정된 버킷에 대한 퍼블릭 액세스 차단 구성을 표시합니다.

```
aws s3api get-public-access-block --bucket my-bucket
```

출력:

```
{
    "PublicAccessBlockConfiguration": {
        "IgnorePublicAcls": true,
        "BlockPublicPolicy": true,
        "BlockPublicAcls": true,
        "RestrictPublicBuckets": true
    }
}
```

- 자세한 API 내용은 명령 참조GetPublicAccessBlock의 섹션을 참조하세요. AWS CLI

### get-sites

다음 코드 예시에서는 get-sites을 사용하는 방법을 보여 줍니다.

AWS CLI

사이트를 가져오려면

다음 get-sites 예제에서는 지정된 글로벌 네트워크의 사이트를 가져옵니다.

```
aws networkmanager get-sites \
    --global-network-id global-network-01231231231231231 \
    --region us-west-2
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetSites의 섹션을 참조하세요. AWS CLI

### get-transit-gateway-registrations

다음 코드 예시에서는 get-transit-gateway-registrations을 사용하는 방법을 보여 줍니다.

AWS CLI

전송 게이트웨이 등록을 가져오려면

다음 get-transit-gateway-registrations 예제에서는 지정된 글로벌 네트워크에 등록된 전송 게이트웨이를 가져옵니다.

```
aws networkmanager get-transit-gateway-registrations \
    --global-network-id global-network-01231231231231231 \
    --region us-west-2
```

출력:

```
{
    "TransitGatewayRegistrations": [
        {
            "GlobalNetworkId": "global-network-01231231231231231",
            "TransitGatewayArn": "arn:aws:ec2:us-west-2:123456789012:transit-
gateway/tgw-123abc05e04123abc",
            "State": {
                "Code": "AVAILABLE"
            }
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetTransitGatewayRegistrations의 섹션을 참조하세요. AWS CLI

### get-vpc-attachment

다음 코드 예시에서는 get-vpc-attachment을 사용하는 방법을 보여 줍니다.

AWS CLI

VPC 첨부 파일을 가져오려면

다음 get-vpc-attachment 예제에서는 VPC 첨부 파일에 대한 정보를 반환합니다.

```
aws networkmanager get-vpc-attachment \
    --attachment-id  attachment-03b7ea450134787da
```

출력:

```
{
    "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
        }
    }
}
```

자세한 내용은 클라우드 WAN 사용 설명서의 첨부 파일을 참조하세요.

- 자세한 API 내용은 명령 참조GetVpcAttachment의 섹션을 참조하세요. AWS CLI

### `list-bucket-analytics-configurations`

다음 코드 예시에서는 `list-bucket-analytics-configurations`을 사용하는 방법을 보여 줍니다.

AWS CLI

버킷의 분석 구성 목록을 검색하려면

다음 `list-bucket-analytics-configurations`는 지정된 버킷에 대한 분석 구성 목록을 검색합니다.

```
aws s3api list-bucket-analytics-configurations \
    --bucket my-bucket
```

출력:

```
{
    "AnalyticsConfigurationList": [
        {
            "StorageClassAnalysis": {},
            "Id": "1"
        }
    ],
    "IsTruncated": false
}
```

- 자세한 API 내용은 명령 참조ListBucketAnalyticsConfigurations의 섹션을 참조하세요. AWS CLI

### `list-bucket-metrics-configurations`

다음 코드 예시에서는 `list-bucket-metrics-configurations`을 사용하는 방법을 보여 줍니다.

AWS CLI

버킷에 대한 지표 구성 목록을 검색하려면

다음 `list-bucket-metrics-configurations` 예제에서는 지정된 버킷에 대한 지표 구성 목록을 검색합니다.

```
aws s3api list-bucket-metrics-configurations \
    --bucket my-bucket
```

출력:

```
{
    "IsTruncated": false,
    "MetricsConfigurationList": [
        {
            "Filter": {
                "Prefix": "logs"
            },
            "Id": "123"
        },
        {
            "Filter": {
                "Prefix": "tmp"
            },
            "Id": "234"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListBucketMetricsConfigurations의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스의 태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 지정된 디바이스 리소스()의 태그를 나열합니다`device-07f6fd08867abc123`.

```
aws networkmanager list-tags-for-resource \
    --resource-arn arn:aws:networkmanager::123456789012:device/global-
network-01231231231231231/device-07f6fd08867abc123 \
    --region us-west-2
```

출력:

```
{
    "TagList": [
        {
            "Key": "Network",
            "Value": "Northeast"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### put-bucket-metrics-configuration

다음 코드 예시에서는 put-bucket-metrics-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

버킷에 대한 지표 구성을 설정하려면

다음 put-bucket-metrics-configuration 예제에서는 지정된 버킷에 대해 ID 123으로 지표 구성을 설정합니다.

```
aws s3api put-bucket-metrics-configuration \
    --bucket my-bucket \
    --id 123 \
    --metrics-configuration '{"Id": "123", "Filter": {"Prefix": "logs"}}'
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조PutBucketMetricsConfiguration의 섹션을 참조하세요. AWS CLI

### put-object-retention

다음 코드 예시에서는 put-object-retention을 사용하는 방법을 보여 줍니다.

AWS CLI

객체의 객체 보존 구성을 설정하는 방법

다음 put-object-retention 예시에서는 지정된 객체에 2025-01-01까지 객체 보존 구성을 설정합니다.

```
aws s3api put-object-retention \
    --bucket my-bucket-with-object-lock \
    --key doc1.rtf \
    --retention '{ "Mode": "GOVERNANCE", "RetainUntilDate": "2025-01-01T00:00:00" }'
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조PutObjectRetention의 섹션을 참조하세요. AWS CLI

### put-public-access-block

다음 코드 예시에서는 put-public-access-block을 사용하는 방법을 보여 줍니다.

AWS CLI

버킷에 대한 퍼블릭 액세스 차단 구성을 설정하려면

다음 put-public-access-block 예제에서는 지정된 버킷에 대한 제한적인 블록 퍼블릭 액세스 구성을 설정합니다.

```
aws s3api put-public-access-block \
    --bucket my-bucket \
    --public-access-block-
configuration "BlockPublicAcls=true,IgnorePublicAcls=true,BlockPublicPolicy=true,RestrictPub
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조PutPublicAccessBlock의 섹션을 참조하세요. AWS CLI

### register-transit-gateway

다음 코드 예시에서는 register-transit-gateway을 사용하는 방법을 보여 줍니다.

AWS CLI

글로벌 네트워크에 전송 게이트웨이를 등록하려면

다음 register-transit-gateway 예제에서는 지정된 글로벌 네트워크에 전송 게이트웨이tgw-123abc05e04123abc를 등록합니다.

```
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
```

출력:

```
{
    "TransitGatewayRegistration": {
        "GlobalNetworkId": "global-network-01231231231231231",
        "TransitGatewayArn": "arn:aws:ec2:us-west-2:123456789012:transit-gateway/
tgw-123abc05e04123abc",
        "State": {
            "Code": "PENDING"
        }
    }
}
```

자세한 내용은 Transit Gateway Network Manager 가이드의 Transit Gateway 등록을 참조하세요.

- 자세한 API 내용은 명령 참조RegisterTransitGateway의 섹션을 참조하세요. AWS CLI

### reject-attachment

다음 코드 예시에서는 reject-attachment을 사용하는 방법을 보여 줍니다.

AWS CLI

첨부 파일을 거부하려면

다음 reject-attachment 예제에서는 VPC 연결 요청을 거부합니다.

```
aws networkmanager reject-attachment \
    --attachment-id  attachment-03b7ea450134787da
```

출력:

```
{
```

```
    "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"
    }
}
```

자세한 내용은 클라우드 WAN 사용 설명서의 첨부 파일 수락을 참조하세요.

- 자세한 API 내용은 명령 참조RejectAttachment의 섹션을 참조하세요. AWS CLI

### start-route-analysis

다음 코드 예시에서는 start-route-analysis을 사용하는 방법을 보여 줍니다.

AWS CLI

라우팅 분석을 시작하려면

다음 start-route-analysis 예제에서는 선택적 를 포함하여 소스와 대상 간의 분석을 시작합니다include-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
```

출력:

```
{
    "RouteAnalysis": {
        "GlobalNetworkId": "global-network-00aa0aaa0b0aaa000
        "OwnerAccountId": "1111222233333",
```

```
        "RouteAnalysisId": "a1873de1-273c-470c-1a2bc2345678",
        "StartTimestamp": 1695760154.0,
        "Status": "RUNNING",
        "Source": {
            "TransitGatewayAttachmentArn": "arn:aws:ec2:us-
east-1:111122223333:transit-gateway-attachment/tgw-attach-1234567890abcdef0,
            "TransitGatewayArn": "arn:aws:ec2:us-east-1:111122223333:transit-
gateway/tgw-abcdef01234567890",
            "IpAddress": "10.0.0.0"
        },
        "Destination": {
            "TransitGatewayAttachmentArn": "arn:aws:ec2:us-
west-1:555555555555:transit-gateway-attachment/tgw-attach-021345abcdef6789",
            "TransitGatewayArn": "arn:aws:ec2:us-west-1:111122223333:transit-
gateway/tgw-09876543210fedcba0",
            "IpAddress": "11.0.0.0"
        },
        "IncludeReturnPath": true,
        "UseMiddleboxes": false
    }
}
```

자세한 내용은 Transit Gateways용 글로벌 네트워크 사용 설명서의 Route Analyzer를 참조하세요. AWS

- 자세한 API 내용은 명령 참조StartRouteAnalysis의 섹션을 참조하세요. AWS CLI

### `tag-resource`

다음 코드 예시에서는 `tag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 태그를 적용하려면

다음 `tag-resource` 예제에서는 태그를 디바이스 `Network=Northeast`에 적용합니다`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
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### untag-resource

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에서 태그를 제거하려면

다음 untag-resource 예제에서는 디바이스 Network에서 키가 있는 태그를 제거합니다device-07f6fd08867abc123.

```
aws networkmanager untag-resource \
    --resource-arn arn:aws:networkmanager::123456789012:device/global-network-01231231231231231/device-07f6fd08867abc123 ]
    --tag-keys Network \
    --region us-west-2
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-device

다음 코드 예시에서는 update-device을 사용하는 방법을 보여 줍니다.

AWS CLI

디바이스를 업데이트하려면

다음 update-device 예제에서는 디바이스의 사이트 ID를 지정device-07f6fd08867abc123하여 디바이스를 업데이트합니다.

```
aws networkmanager update-device \
    --global-network-id global-network-01231231231231231 \
    --device-id device-07f6fd08867abc123 \
    --site-id site-444555aaabbb11223 \
    --region us-west-2
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Transit Gateway Network Manager 가이드의 디바이스 작업을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateDevice의 섹션을 참조하세요. AWS CLI

### update-global-network

다음 코드 예시에서는 update-global-network을 사용하는 방법을 보여 줍니다.

AWS CLI

글로벌 네트워크를 업데이트하려면

다음 update-global-network 예제에서는 글로벌 네트워크 에 대한 설명을 업데이트합니다global-network-01231231231231231.

```
aws networkmanager update-global-network \
    --global-network-id global-network-01231231231231231 \
    --description "Head offices" \
    --region us-west-2
```

출력:

```
{
    "GlobalNetwork": {
        "GlobalNetworkId": "global-network-01231231231231231",
```

```
        "GlobalNetworkArn": "arn:aws:networkmanager::123456789012:global-network/
global-network-01231231231231231",
        "Description": "Head offices",
        "CreatedAt": 1575553525.0,
        "State": "UPDATING"
    }
}
```

자세한 내용은 Transit Gateway Network Manager 가이드의 글로벌 네트워크를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateGlobalNetwork의 섹션을 참조하세요. AWS CLI

### update-link

다음 코드 예시에서는 update-link을 사용하는 방법을 보여 줍니다.

AWS CLI

링크를 업데이트하려면

다음 update-link 예제에서는 링크 에 대한 대역폭 정보를 업데이트합니다link-11112222aaaabbbb1.

```
aws networkmanager update-link \
    --global-network-id global-network-01231231231231231 \
    --link-id link-11112222aaaabbbb1 \
    --bandwidth  UploadSpeed=20,DownloadSpeed=20 \
    --region us-west-2
```

출력:

```
{
    "Link": {
        "LinkId": "link-11112222aaaabbbb1",
        "LinkArn": "arn:aws:networkmanager::123456789012:link/global-
network-01231231231231231/link-11112222aaaabbbb1",
        "GlobalNetworkId": "global-network-01231231231231231",
        "SiteId": "site-444555aaabbb11223",
        "Description": "VPN Link",
        "Type": "broadband",
        "Bandwidth": {
            "UploadSpeed": 20,
            "DownloadSpeed": 20
```

```
        },
        "Provider": "AnyCompany",
        "CreatedAt": 1575555811.0,
        "State": "UPDATING"
    }
}
```

자세한 내용은 Transit Gateway Network Manager 가이드의 링크 작업을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateLink의 섹션을 참조하세요. AWS CLI

### update-site

다음 코드 예시에서는 update-site을 사용하는 방법을 보여 줍니다.

AWS CLI

사이트를 업데이트하려면

다음 update-site 예제에서는 지정된 글로벌 네트워크의 사이트에 대한 설명을 업데이트site-444555aaabbb11223합니다.

```
aws networkmanager update-site \
    --global-network-id global-network-01231231231231231 \
    --site-id site-444555aaabbb11223 \
    --description "New York Office site" \
    --region us-west-2
```

출력:

```
{
    "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"
```

```
    }
}
```

자세한 내용은 Transit Gateway Network Manager 가이드의 사이트 작업을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateSite의 섹션을 참조하세요. AWS CLI

## 를 사용한 Nimble Studio 예제 AWS CLI

다음 코드 예제에서는 Nimble Studio AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `get-eula`

다음 코드 예시에서는 `get-eula`을 사용하는 방법을 보여 줍니다.

AWS CLI

스튜디오에 대한 정보를 얻으려면

다음 `get-eula` 예제에서는 에 대한 정보를 나열합니다EULA.

```
aws nimble get-eula \
    --eula-id "EULAid"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Nimble Studio 사용 설명서의 수락EULA을 참조하세요.

- 자세한 API 내용은 명령 참조GetEula의 섹션을 참조하세요. AWS CLI

### get-launch-profile-details

다음 코드 예시에서는 get-launch-profile-details을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 위젯을 나열하려면

다음 get-launch-profile-details 예제에서는 시작 프로파일에 대한 세부 정보를 나열합니다.

```
aws nimble get-launch-profile-details \
    --studio-id "StudioID" \
    --launch-profile-id "LaunchProfileID"
```

출력:

```
{
    "launchProfile": {
        "arn": "arn:aws:nimble:us-west-2:123456789012:launch-profile/
yeG7lDwNQEiwNTRT7DrV7Q",
        "createdAt": "2022-01-27T21:18:59+00:00",
        "createdBy": "AROA3OO2NEHCCYRNDDIFT: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_-RaAReSOjAnLakbX7Q",
            "vQ5w_TbIRayPkAZgcbyYRA",
            "ZQuMxN99Qfa_Js6ma9TwdA",
            "45KjOSPPRzK2OyvpCuQ6qw"
        ],
        "tags": {
            "resourceArn": "arn:aws:nimble:us-west-2:123456789012:launch-profile/
yeG7lDwNQEiwNTRT7DrV7Q"
        },
        "updatedAt": "2022-01-27T21:19:13+00:00",
        "updatedBy": "AROA3OO2NEHCCYRNDDIFT: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",
                "ggK2eIw6RQyt8PIeeOlD3g",
                "a-D9Wc0VQCKUfxAinCDxaw",
                "RvoNmVXiSrS4LhLTb6ybkw",
                "wtp85BcSTa2NZeNRnMKdjw",
                "Rl-J0fM5Sl2hyIiwWIV6hw"
            ],
            "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",
```

```
                "ggK2eIw6RQyt8PIeeOlD3g",
                "a-D9Wc0VQCKUfxAinCDxaw",
                "RvoNmVXiSrS4LhLTb6ybkw",
                "wtp85BcSTa2NZeNRnMKdjw",
                "Rl-J0fM5Sl2hyIiwWIV6hw"
            ],
            "name": "NimbleStudioLinuxStreamImage",
            "owner": "amazon",
            "platform": "LINUX",
            "state": "READY",
            "streamingImageId": "YGXAqgoWTnCNSV8VP20sHQ",
            "tags": {
                "resourceArn": "arn:aws:nimble:us-west-2:123456789012:streaming-
image/YGXAqgoWTnCNSV8VP20sHQ"
            }
        }
    ],
    "studioComponentSummaries": [
        {
            "description": "FSx for Windows",
            "name": "FSxWindows",
            "studioComponentId": "ZQuMxN99Qfa_Js6ma9TwdA",
            "subtype": "AMAZON_FSX_FOR_WINDOWS",
            "type": "SHARED_FILE_SYSTEM"
        },
        {
            "description": "Instance configuration studio component.",
            "name": "InstanceConfiguration",
            "studioComponentId": "vQ5w_TbIRayPkAZgcbyYRA",
            "subtype": "CUSTOM",
            "type": "CUSTOM"
        },
        {
            "name": "ActiveDirectory",
            "studioComponentId": "_hR_-RaAReSOjAnLakbX7Q",
            "subtype": "AWS_MANAGED_MICROSOFT_AD",
            "type": "ACTIVE_DIRECTORY"
        },
        {
            "description": "Render farm running Deadline",
            "name": "RenderFarm",
            "studioComponentId": "45KjOSPPRzK2OyvpCuQ6qw",
            "subtype": "CUSTOM",
            "type": "COMPUTE_FARM"
```

```
        }
    ]
}
```

자세한 내용은 Amazon Nimble Studio 사용 설명서의 시작 프로파일 생성을 참조하세요.

- 자세한 API 내용은 명령 참조GetLaunchProfileDetails의 섹션을 참조하세요. AWS CLI

### `get-launch-profile`

다음 코드 예시에서는 `get-launch-profile`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 위젯을 나열하려면

다음 `get-launch-profile` 예제에서는 시작 프로파일에 대한 정보를 나열합니다.

```
aws nimble get-launch-profile \
    --studio-id "StudioID" \
    --launch-profile-id "LaunchProfileID"
```

출력:

```
{
    "launchProfile": {
        "arn": "arn:aws:nimble:us-west-2:123456789012:launch-profile/
yeG7lDwNQEiwNTRT7DrV7Q",
        "createdAt": "2022-01-27T21:18:59+00:00",
        "createdBy": "AROA3OO2NEHCCYRNDDIFT: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_-RaAReSOjAnLakbX7Q",
            "vQ5w_TbIRayPkAZgcbyYRA",
            "ZQuMxN99Qfa_Js6ma9TwdA",
            "45KjOSPPRzK2OyvpCuQ6qw"
        ],
        "tags": {},
        "updatedAt": "2022-01-27T21:19:13+00:00",
        "updatedBy": "AROA3OO2NEHCCYRNDDIFT: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"
            }
        ]
    }
}
```

자세한 내용은 Amazon Nimble Studio 사용 설명서의 시작 프로파일 생성을 참조하세요.

- 자세한 API 내용은 명령 참조GetLaunchProfile의 섹션을 참조하세요. AWS CLI

### get-studio

다음 코드 예시에서는 get-studio을 사용하는 방법을 보여 줍니다.

AWS CLI

스튜디오에 대한 정보를 얻으려면

다음 get-studio 예제에서는 AWS 계정의 스튜디오를 나열합니다.

```
aws nimble get-studio \
    --studio-id "StudioID"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Nimble Studio 사용 설명서의 Amazon Nimble Studio란 무엇입니까?를 참조하세요.

- 자세한 API 내용은 명령 참조GetStudio의 섹션을 참조하세요. AWS CLI

### `list-eula-acceptances`

다음 코드 예시에서는 `list-eula-acceptances`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 위젯을 나열하려면

다음 `list-eula-acceptances` 예제에서는 AWS 계정EULAs에서 허용되는 를 나열합니다.

```
aws nimble list-eula-acceptances \
    --studio-id "StudioID"
```

출력:

```
{
    "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-J0fM5Sl2hyIiwWIV6hw"
        },
        {
            "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": "ovO87PnhQ4-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": "5YeXje4yROamuTESGvqIAQ",
            "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": "ggK2eIw6RQyt8PIeeOlD3g"
        },
        {
            "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"
        }
    ]
}
```

자세한 내용은 Amazon Nimble Studio 사용 설명서의 수락EULA을 참조하세요.

- 자세한 API 내용은 명령 참조ListEulaAcceptances의 섹션을 참조하세요. AWS CLI

### `list-eulas`

다음 코드 예시에서는 `list-eulas`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 위젯을 나열하려면

다음 `list-eulas` 예제에서는 AWS 계정EULAs의 를 나열합니다.

```
aws nimble list-eulas
```

출력:

```
{
    "eulas": [
        {
            "content": "https://www.mozilla.org/en-US/MPL/2.0/",
            "createdAt": "2021-04-20T16:45:23+00:00",
            "eulaId": "gJZLygd-Srq_5NNbSfiaLg",
            "name": "Mozilla-FireFox",
            "updatedAt": "2021-04-20T16:45:23+00:00"
        },
        {
            "content": "https://www.awsthinkbox.com/end-user-license-agreement",
            "createdAt": "2021-04-20T16:45:24+00:00",
            "eulaId": "RvoNmVXiSrS4LhLTb6ybkw",
            "name": "Thinkbox-Deadline",
            "updatedAt": "2021-04-20T16:45:24+00:00"
        },
        {
            "content": "https://www.videolan.org/legal.html",
            "createdAt": "2021-04-20T16:45:24+00:00",
            "eulaId": "Rl-J0fM5Sl2hyIiwWIV6hw",
            "name": "Videolan-VLC",
            "updatedAt": "2021-04-20T16:45:24+00:00"
        },
        {
            "content": "https://code.visualstudio.com/license",
            "createdAt": "2021-04-20T16:45:23+00:00",
            "eulaId": "ggK2eIw6RQyt8PIeeOlD3g",
            "name": "Microsoft-VSCode",
            "updatedAt": "2021-04-20T16:45:23+00:00"
        },
        {
            "content": "https://darbyjohnston.github.io/DJV/legal.html#License",
            "createdAt": "2021-04-20T16:45:23+00:00",
            "eulaId": "wtp85BcSTa2NZeNRnMKdjw",
            "name": "DJV-DJV",
            "updatedAt": "2021-04-20T16:45:23+00:00"
        },
```

```
        {
            "content": "https://www.sidefx.com/legal/license-agreement/",
            "createdAt": "2021-04-20T16:45:24+00:00",
            "eulaId": "uu2VDLo-QJeIGWWLBae_UA",
            "name": "SideFX-Houdini",
            "updatedAt": "2021-04-20T16:45:24+00:00"
        },
        {
            "content": "https://www.chaosgroup.com/eula",
            "createdAt": "2021-04-20T16:45:23+00:00",
            "eulaId": "L0HS4P3CRYKVXc2J2LO7Vw",
            "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"
        }
    ]
}
```

자세한 내용은 Amazon Nimble Studio 사용 설명서의 수락EULA을 참조하세요.

- 자세한 API 내용은 명령 참조ListEulas의 섹션을 참조하세요. AWS CLI

### `list-launch-profiles`

다음 코드 예시에서는 `list-launch-profiles`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 위젯을 나열하려면

다음 `list-launch-profiles` 예제에서는 AWS 계정의 시작 프로파일을 나열합니다.

```
aws nimble list-launch-profiles \
    --studio-id "StudioID"
```

출력:

```
{
    "launchProfiles": [
        {
            "arn": "arn:aws:nimble:us-west-2:123456789012:launch-profile/
yeG7lDwNQEiwNTRT7DrV7Q",
            "createdAt": "2022-01-27T21:18:59+00:00",
            "createdBy": "AROA3OO2NEHCCYRNDDIFT: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_-RaAReSOjAnLakbX7Q",
                "vQ5w_TbIRayPkAZgcbyYRA",
                "ZQuMxN99Qfa_Js6ma9TwdA",
                "45KjOSPPRzK2OyvpCuQ6qw"
```

```
            ],
            "tags": {},
            "updatedAt": "2022-01-27T21:19:13+00:00",
            "updatedBy": "AROA3OO2NEHCCYRNDDIFT: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": "AROA3OO2NEHCCYRNDDIFT: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_-RaAReSOjAnLakbX7Q",
                "vQ5w_TbIRayPkAZgcbyYRA",
                "ZQuMxN99Qfa_Js6ma9TwdA",
                "yJSbsHXAQYWk9FXLNusXlQ",
                "45KjOSPPRzK2OyvpCuQ6qw"
            ],
            "tags": {},
            "updatedAt": "2022-01-27T21:19:40+00:00",
            "updatedBy": "AROA3OO2NEHCCYRNDDIFT: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"
                }
            ]
        }
    ]
}
```

자세한 내용은 Amazon Nimble Studio 사용 설명서의 시작 프로파일 생성을 참조하세요.

- 자세한 API 내용은 명령 참조ListLaunchProfiles의 섹션을 참조하세요. AWS CLI

### `list-studio-components`

다음 코드 예시에서는 `list-studio-components`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 위젯을 나열하려면

다음 `list-studio-components` 예제에서는 AWS 계정의 스튜디오 구성 요소를 나열합니다.

```
aws nimble list-studio-components \
    --studio-id "StudioID"
```

출력:

```
{
    "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": "AROA3OO2NEHCCYRNDDIFT: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": "AROA3OO2NEHCCYRNDDIFT:i-EXAMPLE11111"
        },
    ...
}
```

자세한 내용은 Amazon Nimble Studio 사용 설명서의 Amazon Nimble Studio에서 StudioBuilder 를 사용하는 방법을 참조하세요.

- 자세한 API 내용은 명령 참조ListStudioComponents의 섹션을 참조하세요. AWS CLI

### `list-studio-members`

다음 코드 예시에서는 `list-studio-members`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 위젯을 나열하려면

다음 `list-studio-members` 예제에서는 AWS 계정에서 사용 가능한 스튜디오 멤버를 나열합니다.

```
aws nimble list-studio-members \
    --studio-id "StudioID"
```

출력:

```
{
    "members": [
        {
            "identityStoreId": "d-EXAMPLE11111",
            "persona": "ADMINISTRATOR",
            "principalId": "EXAMPLE11111-e9fd012a-94ad-4f16-9866-c69a63ab6486"
        }
    ]
}
```

자세한 내용은 Amazon Nimble Studio 사용 설명서의 스튜디오 사용자 추가를 참조하세요.

- 자세한 API 내용은 명령 참조ListStudioMembers의 섹션을 참조하세요. AWS CLI

### `list-studios`

다음 코드 예시에서는 `list-studios`을 사용하는 방법을 보여 줍니다.

AWS CLI

스튜디오를 나열하려면

다음 `list-studios` 예제에서는 AWS 계정의 스튜디오를 나열합니다.

```
aws nimble list-studios
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Nimble Studio 사용 설명서의 Amazon Nimble Studio란 무엇입니까?를 참조하세요.

- 자세한 API 내용은 명령 참조ListStudios의 섹션을 참조하세요. AWS CLI

## OpenSearch 를 사용한 서비스 예제 AWS CLI

다음 코드 예제에서는 OpenSearch 서비스 AWS Command Line Interface 에서 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `create-elasticsearch-domain`

다음 코드 예시에서는 `create-elasticsearch-domain`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon Elasticsearch Service 도메인을 생성하려면

다음 `create-elasticsearch-domain` 명령은 내에 새 Amazon Elasticsearch Service 도메인을 생성하고 단일 사용자에 대한 액세스를 VPC 제한합니다. Amazon ES는 지정된 서브넷 및 보안 그룹 에서 VPC ID를 추론합니다IDs.

```
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
```

출력:

```
{
    "DomainStatus": {
        "ElasticsearchClusterConfig": {
            "DedicatedMasterEnabled": false,
            "InstanceCount": 1,
            "ZoneAwarenessEnabled": false,
            "InstanceType": "m4.large.elasticsearch"
        },
        "DomainId": "123456789012/vpc-cli-example",
        "CognitoOptions": {
            "Enabled": false
        },
        "VPCOptions": {
            "SubnetIds": [
                "subnet-1a2a3a4a"
            ],
            "VPCId": "vpc-3a4a5a6a",
            "SecurityGroupIds": [
                "sg-2a3a4a5a"
            ],
            "AvailabilityZones": [
                "us-west-1c"
            ]
        },
        "Created": true,
        "Deleted": false,
```

```
        "EBSOptions": {
            "VolumeSize": 10,
            "VolumeType": "standard",
            "EBSEnabled": true
        },
        "Processing": true,
        "DomainName": "vpc-cli-example",
        "SnapshotOptions": {
            "AutomatedSnapshotStartHour": 0
        },
        "ElasticsearchVersion": "6.2",
        "AccessPolicies": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":
\"Allow\",\"Principal\":{\"AWS\":\"arn:aws:iam::123456789012:root\"},\"Action\":
\"es:*\",\"Resource\":\"arn:aws:es:us-west-1:123456789012:domain/vpc-cli-example/*
\"}]}",
        "AdvancedOptions": {
            "rest.action.multi.allow_explicit_index": "true"
        },
        "EncryptionAtRestOptions": {
            "Enabled": false
        },
        "ARN": "arn:aws:es:us-west-1:123456789012:domain/vpc-cli-example"
    }
}
```

자세한 내용은 Amazon Elasticsearch Service 개발자 안내서의 Amazon Elasticsearch Service 도메인 생성 및 관리를 참조하세요. Amazon Elasticsearch Service

- 자세한 API 내용은 명령 참조CreateElasticsearchDomain의 섹션을 참조하세요. AWS CLI

### describe-elasticsearch-domain-config

다음 코드 예시에서는 describe-elasticsearch-domain-config을 사용하는 방법을 보여 줍니다.

AWS CLI

도메인 구성 세부 정보를 가져오려면

다음 describe-elasticsearch-domain-config 예제에서는 각 개별 도메인 구성 요소에 대한 상태 정보와 함께 지정된 도메인에 대한 구성 세부 정보를 제공합니다.

```
aws es describe-elasticsearch-domain-config \
```

```
    --domain-name cli-example
```

출력:

```
{
    "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
            }
        }
    }
}
```

자세한 내용은 Amazon Elasticsearch Service 개발자 안내서의 Amazon Elasticsearch Service 도메인 생성 및 관리를 참조하세요. Amazon Elasticsearch Service

- 자세한 API 내용은 명령 참조DescribeElasticsearchDomainConfig의 섹션을 참조하세요. AWS CLI

### describe-elasticsearch-domain

다음 코드 예시에서는 describe-elasticsearch-domain을 사용하는 방법을 보여 줍니다.

AWS CLI

단일 도메인에 대한 세부 정보를 가져오려면

다음 describe-elasticsearch-domain 예제에서는 지정된 도메인에 대한 구성 세부 정보를 제공합니다.

```
aws es describe-elasticsearch-domain \
    --domain-name cli-example
```

출력:

```
{
    "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
        }
```

```
    }
}
```

자세한 내용은 Amazon Elasticsearch Service 개발자 안내서의 Amazon Elasticsearch Service 도메인 생성 및 관리를 참조하세요. Amazon Elasticsearch Service

- 자세한 API 내용은 명령 참조DescribeElasticsearchDomain의 섹션을 참조하세요. AWS CLI

### `describe-elasticsearch-domains`

다음 코드 예시에서는 `describe-elasticsearch-domains`을 사용하는 방법을 보여 줍니다.

AWS CLI

하나 이상의 도메인에 대한 세부 정보를 가져오려면

다음 `describe-elasticsearch-domains` 예제에서는 하나 이상의 도메인에 대한 구성 세부 정보를 제공합니다.

```
aws es describe-elasticsearch-domains \
    --domain-names cli-example-1 cli-example-2
```

출력:

```
{
    "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
            }
        }
    ]
}
```

자세한 내용은 Amazon Elasticsearch Service 개발자 안내서의 Amazon Elasticsearch Service 도메인 생성 및 관리를 참조하세요. Amazon Elasticsearch Service

- 자세한 API 내용은 명령 참조DescribeElasticsearchDomains의 섹션을 참조하세요. AWS CLI

### describe-reserved-elasticsearch-instances

다음 코드 예시에서는 describe-reserved-elasticsearch-instances을 사용하는 방법을 보여 줍니다.

AWS CLI

예약된 모든 인스턴스를 보려면

다음 describe-elasticsearch-domains 예제에서는 리전에서 예약한 모든 인스턴스에 대한 요약을 제공합니다.

```
aws es describe-reserved-elasticsearch-instances
```

출력:

```
{
```

```
    "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"
    }]
}
```

자세한 내용은 Amazon Elasticsearch Service 개발자 안내서의 예약 인스턴스를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeReservedElasticsearchInstances의 섹션을 참조하세요. AWS CLI

### `list-domain-names`

다음 코드 예시에서는 `list-domain-names`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 도메인을 나열하려면

다음 `list-domain-names` 예제에서는 리전의 모든 도메인에 대한 간략한 요약을 제공합니다.

```
aws es list-domain-names
```

출력:

```
{
    "DomainNames": [{
            "DomainName": "cli-example-1"
```

```
        },
        {
            "DomainName": "cli-example-2"
        }
    ]
}
```

자세한 내용은 Amazon Elasticsearch Service 개발자 안내서의 Amazon Elasticsearch Service 도메인 생성 및 관리를 참조하세요. Amazon Elasticsearch Service

- 자세한 API 내용은 명령 참조ListDomainNames의 섹션을 참조하세요. AWS CLI

## AWS OpsWorks 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS OpsWorks.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `assign-instance`

다음 코드 예시에서는 `assign-instance`을 사용하는 방법을 보여 줍니다.

AWS CLI

계층에 등록된 인스턴스를 할당하려면

다음 예제에서는 등록된 인스턴스를 사용자 지정 계층에 할당합니다.

```
aws opsworks --region us-east-1 assign-instance --instance-id 4d6d1710-ded9-42a1-b08e-b043ad7af1e2 --layer-ids 26cf1d32-6876-42fa-bbf1-9cadc0bff938
```

출력 : 없음.

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 계층에 등록된 인스턴스 할당을 참조하세요.

- 자세한 API 내용은 명령 참조AssignInstance의 섹션을 참조하세요. AWS CLI

### `assign-volume`

다음 코드 예시에서는 `assign-volume`을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스에 등록된 볼륨을 할당하려면

다음 예제에서는 등록된 Amazon Elastic Block Store(Amazon EBS) 볼륨을 인스턴스에 할당합니다. 볼륨은 볼륨 ID로 식별되며, 이는 Amazon Elastic Compute Cloud(Amazon EC2) 볼륨 ID가 아닌 스택에 볼륨을 등록할 때 AWS OpsWorks 할당GUID하는 입니다. 를 실행하기 전에 먼저 를 실행update-volume하여 볼륨에 마운트 포인트를 할당해야 `assign-volume`합니다.

```
aws opsworks --region us-east-1 assign-volume --instance-id 4d6d1710-ded9-42a1-b08e-b043ad7af1e2 --volume-id 26cf1d32-6876-42fa-bbf1-9cadc0bff938
```

출력 : 없음.

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 인스턴스에 Amazon EBS 볼륨 할당을 참조하세요.

- 자세한 API 내용은 명령 참조AssignVolume의 섹션을 참조하세요. AWS CLI

### `associate-elastic-ip`

다음 코드 예시에서는 `associate-elastic-ip`을 사용하는 방법을 보여 줍니다.

AWS CLI

탄력적 IP 주소를 인스턴스와 연결하려면

다음 예제에서는 탄력적 IP 주소를 지정된 인스턴스와 연결합니다.

```
aws opsworks --region us-east-1 associate-elastic-ip --instance-
id dfe18b02-5327-493d-91a4-c5c0c448927f --elastic-ip 54.148.130.96
```

출력 : 없음.

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 리소스 관리를 참조하세요.

- 자세한 API 내용은 명령 참조AssociateElasticIp의 섹션을 참조하세요. AWS CLI

### attach-elastic-load-balancer

다음 코드 예시에서는 attach-elastic-load-balancer을 사용하는 방법을 보여 줍니다.

AWS CLI

로드 밸런서를 계층에 연결하려면

다음 예제에서는 이름으로 식별되는 로드 밸런서를 지정된 계층에 연결합니다.

```
aws opsworks --region us-east-1 attach-elastic-load-balancer --elastic-load-
balancer-name Java-LB --layer-id 888c5645-09a5-4d0e-95a8-812ef1db76a4
```

출력 : 없음.

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 Elastic Load Balancing을 참조하세요.

- 자세한 API 내용은 명령 참조AttachElasticLoadBalancer의 섹션을 참조하세요. AWS CLI

### create-app

다음 코드 예시에서는 create-app을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 앱 생성

다음 예제에서는 GitHub 리포지토리에 저장된 코드implePHPApp 에서 S라는 PHP 앱을 생성합니다. 명령은 애플리케이션 소스 정의의 약식 형식을 사용합니다.

```
aws opsworks create-app \
    --region us-east-1 \
    --stack-id f6673d70-32e6-4425-8999-265dd002fec7 \
    --name SimplePHPApp \
    --type php \
    --app-source Type=git,Url=git://github.com/amazonwebservices/opsworks-demo-php-simple-app.git,Revision=version1
```

출력:

```
{
    "AppId": "6cf5163c-a951-444f-a8f7-3716be75f2a2"
}
```

예제 2: 데이터베이스가 연결된 앱을 생성하려면

다음 예제에서는 퍼블릭 S3 버킷의 .zip 아카이브에 저장된 코드에서 JSP 앱을 생성합니다. DB RDS 인스턴스를 연결하여 앱의 데이터 스토어 역할을 합니다. 애플리케이션 및 데이터베이스 소스는 명령을 실행하는 디렉터리에 있는 별도의 JSON 파일에 정의됩니다.

```
aws opsworks create-app \
    --region us-east-1 \
    --stack-id 8c428b08-a1a1-46ce-a5f8-feddc43771b8 \
    --name SimpleJSP \
    --type java \
    --app-source file://appsource.json \
    --data-sources file://datasource.json
```

애플리케이션 소스 정보는 에 appsource.json 있으며 다음을 포함합니다.

```
{
    "Type": "archive",
    "Url": "https://s3.amazonaws.com/opsworks-demo-assets/simplejsp.zip"
}
```

데이터베이스 소스 정보는 에 datasource.json 있으며 다음을 포함합니다.

```
[
    {
        "Type": "RdsDbInstance",
        "Arn": "arn:aws:rds:us-west-2:123456789012:db:clitestdb",
```

```
        "DatabaseName": "mydb"
    }
]
```

참고 : RDS DB 인스턴스의 경우 먼저 `register-rds-db-instance`를 사용하여 인스턴스를 스택에 등록해야 합니다. MySQL App Server 인스턴스의 경우 를 `Type`로 설정합니다`OpsworksMysqlInstance`. 이러한 인스턴스는 에서 생성 AWS OpsWorks되므로 등록할 필요가 없습니다.

출력:

```
{
    "AppId": "26a61ead-d201-47e3-b55c-2a7c666942f8"
}
```

자세한 내용은 AWS OpsWorks 사용 설명서의 앱 추가를 참조하세요.

- 자세한 API 내용은 명령 참조CreateApp의 섹션을 참조하세요. AWS CLI

### `create-deployment`

다음 코드 예시에서는 `create-deployment`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 앱을 배포하고 스택 명령을 실행하려면

다음 예제에서는 `create-deployment` 명령을 사용하여 앱을 배포하고 스택 명령을 실행하는 방법을 보여줍니다. 명령을 지정하는 JSON 객체의 따옴표(") 문자 앞에는 모두 이스케이프 문자(\)가 표시됩니다. 이스케이프 문자가 없으면 명령이 잘못된 JSON 오류를 반환할 수 있습니다.

다음 `create-deployment` 예제에서는 앱을 지정된 스택에 배포합니다.

```
aws opsworks create-deployment \
    --stack-id cfb7e082-ad1d-4599-8e81-de1c39ab45bf \
    --app-id 307be5c8-d55d-47b5-bd6e-7bd417c6c7eb
    --command "{\"Name\":\"deploy\"}"
```

출력:

```
{
    "DeploymentId": "5746c781-df7f-4c87-84a7-65a119880560"
```

```
}
```

예제 2: Rails 앱을 배포하고 데이터베이스를 마이그레이션하려면

다음 `create-deployment` 명령은 Ruby on Rails 앱을 지정된 스택에 배포하고 데이터베이스를 마이그레이션합니다.

```
aws opsworks create-deployment \
    --stack-id cfb7e082-ad1d-4599-8e81-de1c39ab45bf \
    --app-id 307be5c8-d55d-47b5-bd6e-7bd417c6c7eb \
    --command "{\"Name\":\"deploy\", \"Args\":{\"migrate\":[\"true\"]}}"
```

출력:

```
{
    "DeploymentId": "5746c781-df7f-4c87-84a7-65a119880560"
}
```

배포에 대한 자세한 내용은 AWS OpsWorks 사용 설명서의 앱 배포를 참조하세요.

예제 3: 레시피 실행

다음 `create-deployment` 명령은 지정된 스택의 인스턴스phpapp::appsetup에서 사용자 지정 레시피인 를 실행합니다.

```
aws opsworks create-deployment \
    --stack-id 935450cc-61e0-4b03-a3e0-160ac817d2bb \
    --command "{\"Name\":\"execute_recipes\", \"Args\":{\"recipes\":
[\"phpapp::appsetup\"]}}"
```

출력:

```
{
    "DeploymentId": "5cbaa7b9-4e09-4e53-aa1b-314fbd106038"
}
```

자세한 내용은 AWS OpsWorks 사용 설명서의 스택 명령 실행을 참조하세요.

예제 4: 종속성 설치

다음 `create-deployment` 명령은 지정된 스택의 인스턴스에 패키지 또는 Ruby 보석과 같은 종속성을 설치합니다.

```
aws opsworks create-deployment \
    --stack-id 935450cc-61e0-4b03-a3e0-160ac817d2bb \
    --command "{\"Name\":\"install_dependencies\"}"
```

출력:

```
{
    "DeploymentId": "aef5b255-8604-4928-81b3-9b0187f962ff"
}
```

자세한 내용은 AWS OpsWorks 사용 설명서의 스택 명령 실행을 참조하세요.

- 자세한 API 내용은 명령 참조CreateDeployment의 섹션을 참조하세요. AWS CLI

### `create-instance`

다음 코드 예시에서는 `create-instance`을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스를 생성하려면

다음 `create-instance` 명령은 지정된 스택에 myinstance1이라는 m1.large Amazon Linux 인스턴스를 생성합니다. 인스턴스는 한 계층에 할당됩니다.

```
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"
```

자동 생성된 이름을 사용하려면 를 호출합니다. get-hostname-suggestion이 호출은 스택을 생성할 때 지정한 테마를 기반으로 호스트 이름을 생성합니다. 그런 다음 해당 이름을 호스트 이름 인수에 전달합니다.

출력:

```
{
  "InstanceId": "5f9adeaa-c94c-42c6-aeef-28a5376002cd"
}
```

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 계층에 인스턴스 추가를 참조하세요.

- 자세한 API 내용은 명령 참조CreateInstance의 섹션을 참조하세요. AWS CLI

### create-layer

다음 코드 예시에서는 create-layer을 사용하는 방법을 보여 줍니다.

AWS CLI

계층을 생성하려면

다음 create-layer 명령은 지정된 스택yPHPLayer 에 M이라는 PHP 앱 서버 계층을 생성합니다.

```
aws opsworks create-layer --region us-east-1 --stack-
id f6673d70-32e6-4425-8999-265dd002fec7 --type php-app --name MyPHPLayer --
shortname myphplayer
```

출력:

```
{
  "LayerId": "0b212672-6b4b-40e4-8a34-5a943cf2e07a"
}
```

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 계층 생성 방법을 참조하세요.

- 자세한 API 내용은 명령 참조CreateLayer의 섹션을 참조하세요. AWS CLI

### create-server

다음 코드 예시에서는 create-server을 사용하는 방법을 보여 줍니다.

AWS CLI

서버를 생성하려면

다음 create-server 예제에서는 기본 리전automate-06에 이름이 지정된 새 Chef Automate 서버를 생성합니다. 기본값은 보존할 백업 수, 유지 관리 및 백업 시작 시간과 같은 대부분의 다른 설정에 사용됩니다. create-server 명령을 실행하기 전에 Opsworks AWS OpsWorks for Chef Automate 사용 설명서의 Chef Automate 시작하기에서 사전 조건을 완료합니다. AWS

```
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"
```

출력:

```
{
    "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" ]
    }
}
```

자세한 내용은 for Chef Automate 참조 CreateServer의 섹션을 참조하세요. AWS OpsWorks API

- 자세한 API 내용은 명령 참조CreateServer의 섹션을 참조하세요. AWS CLI

### create-stack

다음 코드 예시에서는 create-stack을 사용하는 방법을 보여 줍니다.

AWS CLI

스택을 생성하려면

다음 create-stack 명령은 스택이라는 CLI 스택을 생성합니다.

```
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
```

service-role-arn 및 default-instance-profile-arn 파라미터가 필요합니다. 일반적으로 첫 번째 스택을 생성할 때 가 AWS OpsWorks 생성하는 를 사용합니다. 계정의 Amazon 리소스 이름(ARNs)을 가져오려면 IAM 콘솔로 이동하여 탐색 패널Roles에서 역할 또는 프로필을 선택하고 Summary 탭을 선택합니다.

출력:

```
{
  "StackId": "f6673d70-32e6-4425-8999-265dd002fec7"
}
```

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 새 스택 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateStack의 섹션을 참조하세요. AWS CLI

### create-user-profile

다음 코드 예시에서는 create-user-profile을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 프로필을 생성하려면

를 호출 create-user-profile하여 사용자 프로필을 생성 AWS OpsWorks 하여 AWS Identity and Access Manager(IAM) 사용자를 로 가져옵니다. 다음 예제에서는 Amazon 리소스 이름()으로 식별되는 IAM 사용자의 사용자 프로필을 cli-user-test 생성합니다ARN. 이 예제에서는 사용자에게 의 SSH 사용자 이름을 할당myusername하고 사용자가 SSH 퍼블릭 키를 지정할 수 있는 자체 관리를 활성화합니다.

```
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
```

출력:

```
{
  "IamUserArn": "arn:aws:iam::123456789102:user/cli-user-test"
}
```

팁 : 이 명령은 연결된 정책에서 부여한 권한을 AWS OpsWorks가진 IAM 사용자를 로 가져옵니다. set-permissions 명령을 사용하여 스택당 AWS OpsWorks 권한을 부여할 수 있습니다.

추가 정보

자세한 내용은 사용 설명서 AWS OpsWorks 의 로 사용자 가져오기를 참조하세요. AWS OpsWorks

- 자세한 API 내용은 명령 참조CreateUserProfile의 섹션을 참조하세요. AWS CLI

### delete-app

다음 코드 예시에서는 delete-app을 사용하는 방법을 보여 줍니다.

AWS CLI

앱을 삭제하려면

다음 예제에서는 지정된 앱을 삭제합니다. 이 앱은 앱 ID로 식별됩니다. AWS OpsWorks 콘솔의 앱 세부 정보 페이지로 이동하거나 describe-apps 명령을 실행하여 앱 ID를 얻을 수 있습니다.

```
aws opsworks delete-app --region us-east-1 --app-id 577943b9-2ec1-4baf-
a7bf-1d347601edc5
```

출력 : 없음.

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 앱을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteApp의 섹션을 참조하세요. AWS CLI

### delete-instance

다음 코드 예시에서는 delete-instance을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스를 삭제하려면

다음 delete-instance 예제에서는 인스턴스 ID로 식별되는 지정된 인스턴스를 삭제합니다. AWS OpsWorks 콘솔에서 인스턴스의 세부 정보 페이지를 열거나 describe-instances 명령을 실행하여 인스턴스 ID를 찾을 수 있습니다.

인스턴스가 온라인 상태인 경우 먼저 를 호출하여 인스턴스를 중지stop-instance한 다음 인스턴스가 중지될 때까지 기다려야 합니다. 를 실행describe-instances하여 인스턴스 상태를 확인합니다.

인스턴스의 Amazon EBS 볼륨 또는 탄력적 IP 주소를 제거하려면 --delete-volumes 또는 --delete-elastic-ip 인수를 각각 추가합니다.

```
aws opsworks delete-instance \
    --region us-east-1 \
    --instance-id 3a21cfac-4a1f-4ce2-a921-b2cfba6f7771
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS OpsWorks 사용 설명서의 AWS OpsWorks 인스턴스 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteInstance의 섹션을 참조하세요. AWS CLI

### delete-layer

다음 코드 예시에서는 delete-layer을 사용하는 방법을 보여 줍니다.

AWS CLI

계층을 삭제하려면

다음 예제에서는 지정된 계층을 삭제합니다. 이 계층은 계층 ID로 식별됩니다. AWS OpsWorks 콘솔에서 계층의 세부 정보 페이지로 이동하거나 `describe-layers` 명령을 실행하여 계층 ID를 얻을 수 있습니다.

참고: 계층을 삭제하기 전에 `delete-instance`를 사용하여 계층의 모든 인스턴스를 삭제해야 합니다.

```
aws opsworks delete-layer --region us-east-1 --layer-id a919454e-b816-4598-
b29a-5796afb498ed
```

출력 : 없음.

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 AWS OpsWorks 인스턴스 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteLayer의 섹션을 참조하세요. AWS CLI

### `delete-stack`

다음 코드 예시에서는 `delete-stack`을 사용하는 방법을 보여 줍니다.

AWS CLI

스택을 삭제하려면

다음 예제에서는 스택 ID로 식별되는 지정된 스택을 삭제합니다. AWS OpsWorks 콘솔에서 스택 설정을 클릭하거나 명령을 실행하여 스택 ID를 얻을 수 있습니다. `describe-stacks`

참고: 계층을 삭제하기 전에 , `delete-instance`및 `delete-appdelete-layer`를 사용하여 스택의 모든 앱, 인스턴스 및 계층을 삭제해야 합니다.

```
aws opsworks delete-stack --region us-east-1 --stack-
id 154a9d89-7e9e-433b-8de8-617e53756c84
```

출력 : 없음.

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 스택 종료를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteStack의 섹션을 참조하세요. AWS CLI

### delete-user-profile

다음 코드 예시에서는 delete-user-profile을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 프로필을 삭제하고 에서 IAM 사용자를 제거하려면 AWS OpsWorks

다음 예제에서는 Amazon 리소스 이름(IAM)으로 식별되는 지정된 AWS Identity and Access Management() 사용자의 사용자 프로파일을 삭제합니다ARN. 작업은 에서 사용자를 제거 AWS OpsWorks하지만 IAM 사용자를 삭제하지는 않습니다. API 해당 작업에 IAM 콘솔, CLI또는 를 사용해야 합니다.

```
aws opsworks --region us-east-1 delete-user-profile --iam-user-
arn arn:aws:iam::123456789102:user/cli-user-test
```

출력 : 없음.

추가 정보

자세한 내용은 사용 설명서 AWS OpsWorks 의 로 사용자 가져오기를 참조하세요. AWS OpsWorks

- 자세한 API 내용은 명령 참조DeleteUserProfile의 섹션을 참조하세요. AWS CLI

### deregister-elastic-ip

다음 코드 예시에서는 deregister-elastic-ip을 사용하는 방법을 보여 줍니다.

AWS CLI

스택에서 탄력적 IP 주소 등록을 취소하려면

다음 예제에서는 스택에서 IP 주소로 식별되는 탄력적 IP 주소의 등록을 취소합니다.

```
aws opsworks deregister-elastic-ip --region us-east-1 --elastic-ip 54.148.130.96
```

출력 : 없음.

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 탄력적 IP 주소 등록 취소를 참조하세요.

- 자세한 API 내용은 명령 참조DeregisterElasticIp의 섹션을 참조하세요. AWS CLI

### deregister-instance

다음 코드 예시에서는 deregister-instance을 사용하는 방법을 보여 줍니다.

AWS CLI

스택에서 등록된 인스턴스를 등록 취소하려면

다음 deregister-instance 명령은 스택에서 등록된 인스턴스의 등록을 취소합니다.

```
aws opsworks --region us-east-1 deregister-instance --instance-id 4d6d1710-
ded9-42a1-b08e-b043ad7af1e2
```

출력 : 없음.

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 등록된 인스턴스 등록 취소를 참조하세요.

- 자세한 API 내용은 명령 참조DeregisterInstance의 섹션을 참조하세요. AWS CLI

### deregister-rds-db-instance

다음 코드 예시에서는 deregister-rds-db-instance을 사용하는 방법을 보여 줍니다.

AWS CLI

스택에서 Amazon RDS DB 인스턴스 등록을 취소하려면

다음 예제에서는 스택ARN에서 로 식별되는 RDS DB 인스턴스의 등록을 취소합니다.

```
aws opsworks deregister-rds-db-instance --region us-east-1 --rds-db-instance-
arn arn:aws:rds:us-west-2:123456789012:db:clitestdb
```

출력 : 없음.

추가 정보

자세한 내용은 ASW OpsWorks 사용 설명서의 Amazon RDS 인스턴스 등록 취소를 참조하세요.

인스턴스 ID: clitestdb 마스터 사용자 이름: cliuser Master PWD: some23!pwd DB 이름: mydb aws opsworks deregister-rds-db-instance --region us-east-1 --rds-db-instance-arn arn:aws:rds:us-west-2:645732743964:db:clitestdb

- 자세한 API 내용은 명령 참조DeregisterRdsDbInstance의 섹션을 참조하세요. AWS CLI

### `deregister-volume`

다음 코드 예시에서는 `deregister-volume`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon EBS 볼륨 등록을 취소하려면

다음 예제에서는 스택에서 EBS 볼륨의 등록을 취소합니다. 볼륨은 볼륨 ID가 아닌 스택에 볼륨을 등록할 때 GUID AWS OpsWorks 할당된 EC2 볼륨 ID로 식별됩니다.

```
aws opsworks deregister-volume --region us-east-1 --volume-id 5c48ef52-3144-4bf5-beaa-fda4deb23d4d
```

출력 : 없음.

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 Amazon EBS 볼륨 등록 취소를 참조하세요.

- 자세한 API 내용은 명령 참조DeregisterVolume의 섹션을 참조하세요. AWS CLI

### `describe-apps`

다음 코드 예시에서는 `describe-apps`을 사용하는 방법을 보여 줍니다.

AWS CLI

앱을 설명하려면

다음 `describe-apps` 명령은 지정된 스택의 앱을 설명합니다.

```
aws opsworks describe-apps \
    --stack-id 38ee91e2-abdc-4208-a107-0b7168b3cc7a \
    --region us-east-1
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS OpsWorks 사용 설명서의 앱을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeApps의 섹션을 참조하세요. AWS CLI

### describe-commands

다음 코드 예시에서는 describe-commands을 사용하는 방법을 보여 줍니다.

AWS CLI

명령을 설명하려면

다음 describe-commands 명령은 지정된 인스턴스의 명령을 설명합니다.

```
aws opsworks describe-commands \
```

```
    --instance-id 8c2673b9-3fe5-420d-9cfa-78d875ee7687 \
    --region us-east-1
```

출력:

```
{
    "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-conten t-type=text%2Fplain",
            "Type": "deploy",
            "CommandId": "899d3d64-0384-47b6-a586-33433aad117c",
            "CreatedAt": "2013-07-25T18:55:29+00:00",
            "ExitCode": 0
        }
    ]
}
```

자세한 내용은 AWS OpsWorks 사용 설명서의 AWS OpsWorks 수명 주기 이벤트를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeCommands의 섹션을 참조하세요. AWS CLI

### describe-deployments

다음 코드 예시에서는 describe-deployments을 사용하는 방법을 보여 줍니다.

AWS CLI

배포를 설명하려면

다음 describe-deployments 명령은 지정된 스택의 배포를 설명합니다.

```
aws opsworks --region us-east-1 describe-deployments --stack-id 38ee91e2-abdc-4208-a107-0b7168b3cc7a
```

출력:

```
{
  "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"
        }
    ]
}
```

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 앱 배포를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeDeployments의 섹션을 참조하세요. AWS CLI

### describe-elastic-ips

다음 코드 예시에서는 describe-elastic-ips을 사용하는 방법을 보여 줍니다.

AWS CLI

탄력적 IP 인스턴스를 설명하려면

다음 describe-elastic-ips 명령은 지정된 인스턴스의 탄력적 IP 주소를 설명합니다.

```
aws opsworks --region us-east-1 describe-elastic-ips --instance-id b62f3e04-
e9eb-436c-a91f-d9e9a396b7b0
```

출력:

```
{
    "ElasticIps": [
        {
            "Ip": "192.0.2.0",
            "Domain": "standard",
            "Region": "us-west-2"
        }
    ]
}
```

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 인스턴스를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeElasticIps의 섹션을 참조하세요. AWS CLI

### `describe-elastic-load-balancers`

다음 코드 예시에서는 `describe-elastic-load-balancers`을 사용하는 방법을 보여 줍니다.

AWS CLI

스택의 탄력적 로드 밸런서를 설명하려면

다음 `describe-elastic-load-balancers` 명령은 지정된 스택의 로드 밸런서를 설명합니다.

```
aws opsworks --region us-west-2 describe-elastic-load-balancers --stack-
id 6f4660e5-37a6-4e42-bfa0-1358ebd9c182
```

출력 : 이 특정 스택에는 하나의 로드 밸런서가 있습니다.

```
{
  "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"
      }
  ]
}
```

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 앱을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeElasticLoadBalancers의 섹션을 참조하세요. AWS CLI

### `describe-instances`

다음 코드 예시에서는 `describe-instances`을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스를 설명하려면

다음 `describe-instances` 명령은 지정된 스택의 인스턴스를 설명합니다.

```
aws opsworks --region us-east-1 describe-instances --stack-id 8c428b08-a1a1-46ce-a5f8-feddc43771b8
```

출력 : 다음 출력 예제는 인스턴스가 두 개 있는 스택에 대한 것입니다. 첫 번째 인스턴스는 등록된 EC2 인스턴스이고 두 번째 인스턴스는 에서 생성되었습니다 AWS OpsWorks.

```
{
  "Instances": [
    {
      "StackId": "71c7ca72-55ae-4b6a-8ee1-a8dcded3fa0f",
      "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-a8dcded3fa0f",
      "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-bbf1-9cadc0bff938"
      ],
      "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"
    }
  ]
}
```

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 인스턴스를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeInstances의 섹션을 참조하세요. AWS CLI

### describe-layers

다음 코드 예시에서는 describe-layers을 사용하는 방법을 보여 줍니다.

AWS CLI

스택의 계층을 설명하려면

다음 describe-layers 명령은 지정된 스택의 계층을 설명합니다.

```
aws opsworks --region us-east-1 describe-layers --stack-id 38ee91e2-abdc-4208-
a107-0b7168b3cc7a
```

출력:

```
{
  "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": []
        }
    ]
}
```

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 계층을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeLayers의 섹션을 참조하세요. AWS CLI

### describe-load-based-auto-scaling

다음 코드 예시에서는 describe-load-based-auto-scaling을 사용하는 방법을 보여 줍니다.

AWS CLI

계층의 로드 기반 조정 구성을 설명하려면

다음 예제에서는 지정된 계층의 로드 기반 조정 구성을 설명합니다. 계층은 계층의 세부 정보 페이지 또는 실행을 통해 찾을 수 있는 계층 ID로 식별됩니다describe-layers.

```
aws opsworks describe-load-based-auto-scaling --region us-east-1 --layer-
ids 6bec29c9-c866-41a0-aba5-fa3e374ce2a1
```

출력 : 예제 계층에는 단일 로드 기반 인스턴스가 있습니다.

```
{
  "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"
    }
  ]
}
```

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 자동 로드 기반 크기 조정 작동 방식을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeLoadBasedAutoScaling의 섹션을 참조하세요. AWS CLI

### describe-my-user-profile

다음 코드 예시에서는 describe-my-user-profile을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 프로필을 가져오려면

다음 예제에서는 명령을 실행하는 AWS Identity and Access Management(IAM) 사용자의 프로파일을 가져오는 방법을 보여줍니다.

```
aws opsworks --region us-east-1 describe-my-user-profile
```

출력 : 간결성을 위해 사용자의 SSH 퍼블릭 키 대부분은 줄임표(...)로 대체됩니다.

```
{
  "UserProfile": {
    "IamUserArn": "arn:aws:iam::123456789012:user/myusername",
    "SshPublicKey": "ssh-rsa AAAAB3NzaC1yc2EAAAABJQ...3LQ4aX9jpxQw== rsa-
key-20141104",
    "Name": "myusername",
    "SshUsername": "myusername"
```

```
  }
}
```

추가 정보

자세한 내용은 사용 설명서 AWS OpsWorks 의 로 사용자 가져오기를 참조하세요. AWS OpsWorks

- 자세한 API 내용은 명령 참조DescribeMyUserProfile의 섹션을 참조하세요. AWS CLI

### `describe-permissions`

다음 코드 예시에서는 `describe-permissions`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자의 스택당 AWS OpsWorks 권한 수준을 얻으려면

다음 예제에서는 지정된 스택에서 AWS Identity and Access Management(IAM) 사용자의 권한 수준을 얻는 방법을 보여줍니다.

```
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
```

출력:

```
{
  "Permissions": [
    {
      "StackId": "d72553d4-8727-448c-9b00-f024f0ba1b06",
      "IamUserArn": "arn:aws:iam::123456789012:user/cli-user-test",
      "Level": "manage",
      "AllowSudo": true,
      "AllowSsh": true
    }
  ]
}
```

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 스택당 권한 수준 부여를 참조하세요.

- 자세한 API 내용은 명령 참조DescribePermissions의 섹션을 참조하세요. AWS CLI

### describe-raid-arrays

다음 코드 예시에서는 describe-raid-arrays을 사용하는 방법을 보여 줍니다.

AWS CLI

RAID 배열을 설명하려면

다음 예제에서는 지정된 스택의 인스턴스에 연결된 RAID 배열을 설명합니다.

```
aws opsworks --region us-east-1 describe-raid-arrays --stack-
id d72553d4-8727-448c-9b00-f024f0ba1b06
```

출력 : 다음은 하나의 RAID 배열이 있는 스택의 출력입니다.

```
{
  "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
    }
  ]
}
```

자세한 내용은 AWS OpsWorks 사용 설명서의 EBS 볼륨을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeRaidArrays의 섹션을 참조하세요. AWS CLI

### describe-rds-db-instances

다음 코드 예시에서는 describe-rds-db-instances을 사용하는 방법을 보여 줍니다.

AWS CLI

스택의 등록된 Amazon RDS 인스턴스를 설명하려면

다음 예제에서는 지정된 스택에 등록된 Amazon RDS 인스턴스를 설명합니다.

```
aws opsworks --region us-east-1 describe-rds-db-instances --stack-
id d72553d4-8727-448c-9b00-f024f0ba1b06
```

출력 : 다음은 하나의 인스턴스가 등록된 스택의 출력입니다RDS.

```
{
  "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.cdlqlk5uwd0k.us-west-2.rds.amazonaws.com",
      "DbUser": "cliuser",
      "DbInstanceIdentifier": "clitestdb"
    }
  ]
}
```

자세한 내용은 AWS OpsWorks 사용 설명서의 리소스 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeRdsDbInstances의 섹션을 참조하세요. AWS CLI

### `describe-stack-provisioning-parameters`

다음 코드 예시에서는 `describe-stack-provisioning-parameters`을 사용하는 방법을 보여 줍니다.

AWS CLI

스택에 대한 프로비저닝 파라미터를 반환하려면

다음 `describe-stack-provisioning-parameters` 예제에서는 지정된 스택에 대한 프로비저닝 파라미터를 반환합니다. 프로비저닝 파라미터에는 가 스택의 인스턴스에서 에이전트를 관리하는 데 OpsWorks 사용하는 에이전트 설치 위치 및 퍼블릭 키와 같은 설정이 포함됩니다.

```
aws opsworks describe-stack-provisioning-parameters \
    --stack-id 62744d97-6faf-4ecb-969b-a086fEXAMPLE
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS OpsWorks 사용 설명서의 스택 명령 실행을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeStackProvisioningParameters의 섹션을 참조하세요. AWS CLI

### describe-stack-summary

다음 코드 예시에서는 describe-stack-summary을 사용하는 방법을 보여 줍니다.

AWS CLI

스택의 구성을 설명하려면

다음 `describe-stack-summary` 명령은 지정된 스택의 구성에 대한 요약을 반환합니다.

```
aws opsworks --region us-east-1 describe-stack-summary --stack-id 8c428b08-a1a1-46ce-a5f8-feddc43771b8
```

출력:

```
{
  "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/"
  }
}
```

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 스택을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeStackSummary의 섹션을 참조하세요. AWS CLI

### `describe-stacks`

다음 코드 예시에서는 `describe-stacks`을 사용하는 방법을 보여 줍니다.

AWS CLI

스택을 설명하려면

다음 `describe-stacks` 명령은 계정 스택을 설명합니다.

```
aws opsworks --region us-east-1 describe-stacks
```

출력:


```
{
  "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
\": \"-Djava.awt.headless=true -Xmx256m\"\n  },\n  \"datasources\": {\n    \"ROOT\":
 \"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"
    }
  ]
}
```

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 스택을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeStacks의 섹션을 참조하세요. AWS CLI

### describe-timebased-auto-scaling

다음 코드 예시에서는 describe-timebased-auto-scaling을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스의 시간 기반 크기 조정 구성을 설명하려면

다음 예제에서는 지정된 인스턴스의 시간 기반 크기 조정 구성을 설명합니다. 인스턴스는 인스턴스의 세부 정보 페이지 또는 실행을 통해 찾을 수 있는 인스턴스 ID로 식별됩니다describe-instances.

```
aws opsworks describe-time-based-auto-scaling --region us-east-1 --instance-
ids 701f2ffe-5d8e-4187-b140-77b75f55de8d
```

출력 : 이 예제에는 단일 시간 기반 인스턴스가 있습니다.

```
{
  "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"
        }
      }
    }
  ]
}
```

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 자동 시간 기반 크기 조정 작동 방식을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeTimebasedAutoScaling의 섹션을 참조하세요. AWS CLI

### describe-user-profiles

다음 코드 예시에서는 describe-user-profiles을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 프로필을 설명하려면

다음 describe-user-profiles 명령은 계정의 사용자 프로필을 설명합니다.

```
aws opsworks --region us-east-1 describe-user-profiles
```

출력:

```
{
  "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"
    }
  ]
}
```

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 AWS OpsWorks 사용자 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeUserProfiles의 섹션을 참조하세요. AWS CLI

### describe-volumes

다음 코드 예시에서는 describe-volumes을 사용하는 방법을 보여 줍니다.

AWS CLI

스택의 볼륨을 설명하려면

다음 예제에서는 스택의 EBS 볼륨을 설명합니다.

```
aws opsworks --region us-east-1 describe-volumes --stack-id 8c428b08-a1a1-46ce-a5f8-feddc43771b8
```

출력:

```
{
  "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
    }
  ]
}
```

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 리소스 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeVolumes의 섹션을 참조하세요. AWS CLI

### `detach-elastic-load-balancer`

다음 코드 예시에서는 `detach-elastic-load-balancer`을 사용하는 방법을 보여 줍니다.

AWS CLI

로드 밸런서를 계층에서 분리하려면

다음 예제에서는 이름으로 식별되는 로드 밸런서를 계층에서 분리합니다.

```
aws opsworks --region us-east-1 detach-elastic-load-balancer --elastic-load-
balancer-name Java-LB --layer-id 888c5645-09a5-4d0e-95a8-812ef1db76a4
```

출력 : 없음.

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 Elastic Load Balancing을 참조하세요.

- 자세한 API 내용은 명령 참조DetachElasticLoadBalancer의 섹션을 참조하세요. AWS CLI

### `disassociate-elastic-ip`

다음 코드 예시에서는 `disassociate-elastic-ip`을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스에서 탄력적 IP 주소 연결을 해제하려면

다음 예제에서는 지정된 인스턴스에서 탄력적 IP 주소의 연결을 해제합니다.

```
aws opsworks --region us-east-1 disassociate-elastic-ip --elastic-ip 54.148.130.96
```

출력 : 없음.

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 리소스 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DisassociateElasticIp의 섹션을 참조하세요. AWS CLI

### get-hostname-suggestion

다음 코드 예시에서는 get-hostname-suggestion을 사용하는 방법을 보여 줍니다.

AWS CLI

계층의 다음 호스트 이름을 가져오려면

다음 예제에서는 지정된 계층에 대해 다음 번 생성된 호스트 이름을 가져옵니다. 이 예제에 사용되는 계층은 인스턴스가 하나 있는 Java Application Server 계층입니다. 스택의 호스트 이름 테마는 기본값인 Layer_Dependent입니다.

```
aws opsworks --region us-east-1 get-hostname-suggestion --layer-
id 888c5645-09a5-4d0e-95a8-812ef1db76a4
```

출력:

```
{
  "Hostname": "java-app2",
  "LayerId": "888c5645-09a5-4d0e-95a8-812ef1db76a4"
}
```

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 새 스택 생성을 참조하세요.

- 자세한 API 내용은 명령 참조GetHostnameSuggestion의 섹션을 참조하세요. AWS CLI

### reboot-instance

다음 코드 예시에서는 reboot-instance을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스를 재부팅하려면

다음 예제에서는 인스턴스를 재부팅합니다.

```
aws opsworks --region us-east-1 reboot-instance --instance-
id dfe18b02-5327-493d-91a4-c5c0c448927f
```

출력 : 없음.

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 인스턴스 재부팅을 참조하세요.

- 자세한 API 내용은 명령 참조RebootInstance의 섹션을 참조하세요. AWS CLI

### register-elastic-ip

다음 코드 예시에서는 register-elastic-ip을 사용하는 방법을 보여 줍니다.

AWS CLI

스택에 탄력적 IP 주소를 등록하려면

다음 예제에서는 IP 주소로 식별되는 탄력적 IP 주소를 지정된 스택에 등록합니다.

참고: 탄력적 IP 주소는 스택과 동일한 리전에 있어야 합니다.

```
aws opsworks register-elastic-ip --region us-east-1 --stack-
id d72553d4-8727-448c-9b00-f024f0ba1b06 --elastic-ip 54.148.130.96
```

출력

```
{
  "ElasticIp": "54.148.130.96"
}
```

추가 정보

자세한 내용은 OpsWorks 사용 설명서의 스택으로 탄력적 IP 주소 등록을 참조하세요.

- 자세한 API 내용은 명령 참조RegisterElasticIp의 섹션을 참조하세요. AWS CLI

### `register-rds-db-instance`

다음 코드 예시에서는 `register-rds-db-instance`을 사용하는 방법을 보여 줍니다.

AWS CLI

스택에 Amazon RDS 인스턴스를 등록하려면

다음 예제에서는 Amazon 리소스 이름(ARN)으로 식별되는 Amazon RDS DB 인스턴스를 지정된 스택에 등록합니다. 또한 인스턴스의 마스터 사용자 이름과 암호도 지정합니다. AWS OpsWorks는 이러한 값을 검증하지 않습니다. 둘 중 하나가 올바르지 않으면 애플리케이션이 데이터베이스에 연결할 수 없습니다.

```
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
```

출력 : 없음.

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 스택으로 Amazon RDS 인스턴스 등록을 참조하세요.

- 자세한 API 내용은 명령 참조RegisterRdsDbInstance의 섹션을 참조하세요. AWS CLI

### `register-volume`

다음 코드 예시에서는 `register-volume`을 사용하는 방법을 보여 줍니다.

AWS CLI

스택에 Amazon EBS 볼륨을 등록하려면

다음 예제에서는 EBS 볼륨 ID로 식별되는 Amazon 볼륨을 지정된 스택에 등록합니다.

```
aws opsworks register-volume --region us-east-1 --stack-id d72553d4-8727-448c-9b00-
f024f0ba1b06 --ec-2-volume-id vol-295c1638
```

출력:

```
{
  "VolumeId": "ee08039c-7cb7-469f-be10-40fb7f0c05e8"
}
```

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 스택으로 Amazon EBS 볼륨 등록을 참조하세요.

- 자세한 API 내용은 명령 참조RegisterVolume의 섹션을 참조하세요. AWS CLI

### `register`

다음 코드 예시에서는 `register`을 사용하는 방법을 보여 줍니다.

AWS CLI

스택에 인스턴스를 등록하려면

다음 예제에서는 AWS Opsworks 외부에서 생성된 스택에 인스턴스를 등록하는 다양한 방법을 보여줍니다. 등록할 인스턴스 또는 별도의 워크스테이션`register`에서 를 실행할 수 있습니다. 자세한 내용은 AWS OpsWorks 사용 설명서의 Amazon EC2 및 온프레미스 인스턴스 등록을 참조하세요.

참고: 간결성을 위해 예제에서는 `region` 인수를 생략합니다.

Amazon EC2 인스턴스를 등록하려면

EC2 인스턴스를 등록하고 있음을 표시하려면 `--infrastructure-class` 인수를 로 설정합니다`ec2`.

다음 예제에서는 별도의 워크스테이션에서 지정된 스택에 EC2 인스턴스를 등록합니다. 인스턴스는 EC2 ID 로 식별됩니다`i-12345678`. 이 예제에서는 워크스테이션의 기본 SSH 사용자 이름을 사용하고 기본 프라이빗 SSH 키와 같이 암호가 필요하지 않은 인증 기술을 사용하여 인스턴스에 로그인하려고 시도합니다. 실패하면 에서 암호를 `register` 쿼리합니다.

```
aws opsworks register --infrastructure-class=ec2 --stack-id 935450cc-61e0-4b03-a3e0-160ac817d2bb i-12345678
```

다음 예제에서는 별도의 워크스테이션에서 지정된 스택에 EC2 인스턴스를 등록합니다. `--ssh-username` 및 `--ssh-private-key` 인수를 사용하여 명령이 인스턴스에 로그인하는 데 사용하는

SSH 사용자 이름과 프라이빗 키 파일을 명시적으로 지정합니다. `ec2-user`는 Amazon Linux 인스턴스의 표준 사용자 이름입니다. Ubuntu 인스턴스`ubuntu`에 를 사용합니다.

```
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
```

다음 예제에서는 `register` 명령을 실행하는 EC2 인스턴스를 등록합니다. 를 사용하여 인스턴스에 로그인SSH하고 인스턴스 ID 또는 호스트 이름 대신 `register --local` 인수로 실행합니다.

```
aws opsworks register --infrastructure-class ec2 --stack-id 935450cc-61e0-4b03-
a3e0-160ac817d2bb --local
```

온프레미스 인스턴스를 등록하려면

온프레미스 인스턴스를 등록하고 있음을 표시하려면 `--infrastructure-class` 인수를 로 설정합니다`on-premises`.

다음 예제에서는 기존 온프레미스 인스턴스를 별도의 워크스테이션에서 지정된 스택에 등록합니다. 인스턴스는 IP 주소 로 식별됩니다`192.0.2.3`. 이 예제에서는 워크스테이션의 기본 SSH 사용자 이름을 사용하고 기본 프라이빗 SSH 키와 같이 암호가 필요하지 않은 인증 기술을 사용하여 인스턴스에 로그인하려고 시도합니다. 실패하면 에서 암호를 `register` 쿼리합니다.

```
aws opsworks register --infrastructure-class on-premises --stack-
id 935450cc-61e0-4b03-a3e0-160ac817d2bb 192.0.2.3
```

다음 예제에서는 별도의 워크스테이션에서 지정된 스택에 온프레미스 인스턴스를 등록합니다. 인스턴스는 호스트 이름인 로 식별됩니다`host1`. `--override-...` 인수는 각각 `webserver1` 호스트 이름 및 `192.0.2.3` 및 `10.0.0.2` 인스턴스의 퍼블릭 및 프라이빗 IP 주소로 표시되도록 직접 AWS OpsWorks 표시됩니다.

```
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
```

다음 예제에서는 별도의 워크스테이션에서 지정된 스택에 온프레미스 인스턴스를 등록합니다. 인스턴스는 IP 주소로 식별됩니다. 는 지정된 SSH 사용자 이름과 프라이빗 키 파일을 사용하여 인스턴스에 `register` 로그인합니다.

```
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
```

다음 예제에서는 기존 온프레미스 인스턴스를 별도의 워크스테이션에서 지정된 스택에 등록합니다. 명령은 SSH 암호와 인스턴스의 IP 주소를 지정하는 사용자 지정 SSH 명령 문자열을 사용하여 인스턴스에 로그인합니다.

```
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"
```

다음 예제에서는 `register` 명령을 실행하는 온프레미스 인스턴스를 등록합니다. 를 사용하여 인스턴스에 로그인SSH하고 인스턴스 ID 또는 호스트 이름 대신 `register --local` 인수로 실행합니다.

```
aws opsworks register --infrastructure-class on-premises --stack-
id 935450cc-61e0-4b03-a3e0-160ac817d2bb --local
```

출력 : 다음은 EC2 인스턴스 등록을 위한 일반적인 출력입니다.

```
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.
```

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 AWS OpsWorks 스택에 인스턴스 등록을 참조하세요.

- API 자세한 내용은 AWS CLI 명령 참조의 등록을 참조하세요.

### set-load-based-auto-scaling

다음 코드 예시에서는 `set-load-based-auto-scaling`을 사용하는 방법을 보여 줍니다.

AWS CLI

계층에 대한 로드 기반 조정 구성을 설정하려면

다음 예제에서는 지정된 계층에 대한 로드 기반 크기 조정을 활성화하고 해당 계층에 대한 구성을 설정합니다. `create-instance` 를 사용하여 계층에 로드 기반 인스턴스를 추가해야 합니다.

```
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
```

이 예제에서는 라는 작업 디렉터리의 별도의 파일에 업스케일링 임계값 설정을 배치하며`upscale.json`, 여기에는 다음이 포함됩니다.

```
{
  "InstanceCount": 2,
  "ThresholdsWaitTime": 3,
  "IgnoreMetricsTime": 3,
  "CpuThreshold": 85,
  "MemoryThreshold": 85,
```

```
  "LoadThreshold": 85
}
```

이 예제에서는 다운스케일링 임계값 설정을 라는 작업 디렉터리의 별도의 파일에 넣습니다. `downscale.json`여기에는 다음이 포함됩니다.

```
{
"InstanceCount": 2,
"ThresholdsWaitTime": 3,
"IgnoreMetricsTime": 3,
"CpuThreshold": 35,
"MemoryThreshold": 30,
"LoadThreshold": 30
}
```

출력 : 없음.

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 자동 로드 기반 크기 조정 사용을 참조하세요.

- 자세한 API 내용은 명령 참조SetLoadBasedAutoScaling의 섹션을 참조하세요. AWS CLI

### `set-permission`

다음 코드 예시에서는 `set-permission`을 사용하는 방법을 보여 줍니다.

AWS CLI

스택당 AWS OpsWorks 권한 수준을 부여하려면

를 호출 AWS OpsWorks 하여 AWS Identity and Access Management(IAM) 사용자를 로 가져오면 연결된 IAM 정책에서 부여한 권한만 `create-user-profile`사용자에게 부여됩니다. 사용자 정책을 수정하여 AWS OpsWorks 권한을 부여할 수 있습니다. 그러나 사용자를 가져온 다음 `set-permission` 명령을 사용하여 사용자에게 액세스가 필요한 각 스택에 대한 표준 권한 수준 중 하나를 부여하는 것이 더 쉬운 경우가 많습니다.

다음 예제에서는 Amazon 리소스 이름()으로 식별되는 사용자에 대해 지정된 스택에 대한 권한을 부여합니다ARN. 이 예제에서는 사용자에게 스택 인스턴스에 대한 sudo 및 SSH 권한을 포함한 권한 관리 수준을 부여합니다.

```
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
```

출력 : 없음.

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 스택당 AWS OpsWorks 사용자 권한 부여를 참조하세요.

- 자세한 API 내용은 명령 참조SetPermission의 섹션을 참조하세요. AWS CLI

### set-time-based-auto-scaling

다음 코드 예시에서는 set-time-based-auto-scaling을 사용하는 방법을 보여 줍니다.

AWS CLI

계층에 대한 시간 기반 조정 구성을 설정하려면

다음 예제에서는 지정된 인스턴스에 대한 시간 기반 구성을 설정합니다. 먼저 create-instance를 사용하여 인스턴스를 계층에 추가해야 합니다.

```
aws opsworks --region us-east-1 set-time-based-auto-scaling --instance-id 69b6237c-08c0-4edb-a6af-78f3d01cedf2 --auto-scaling-schedule file://schedule.json
```

이 예제에서는 라는 작업 디렉터리의 별도의 파일에 일정을 넣습니다schedule.json. 이 예제에서는 인스턴스가 월요일과 화요일의 정오UTC(협정 세계시)쯤에 몇 시간 동안 켜져 있습니다.

```
{
  "Monday": {
    "10": "on",
    "11": "on",
    "12": "on",
    "13": "on"
  },
  "Tuesday": {
    "10": "on",
    "11": "on",
    "12": "on",
```

```
    "13": "on"
  }
}
```

출력 : 없음.

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 자동 시간 기반 크기 조정 사용을 참조하세요.

- 자세한 API 내용은 명령 참조SetTimeBasedAutoScaling의 섹션을 참조하세요. AWS CLI

### `start-instance`

다음 코드 예시에서는 `start-instance`을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스를 시작하려면

다음 `start-instance` 명령은 지정된 연중무휴 인스턴스를 시작합니다.

```
aws opsworks start-instance --instance-id f705ee48-9000-4890-8bd3-20eb05825aaf
```

출력 : 없음. describe-instances를 사용하여 인스턴스의 상태를 확인합니다.

팁 start-stack을 호출하여 하나의 명령으로 스택의 모든 오프라인 인스턴스를 시작할 수 있습니다.

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 연중무휴 인스턴스 수동 시작, 중지 및 재부팅을 참조하세요.

- 자세한 API 내용은 명령 참조StartInstance의 섹션을 참조하세요. AWS CLI

### `start-stack`

다음 코드 예시에서는 `start-stack`을 사용하는 방법을 보여 줍니다.

AWS CLI

스택의 인스턴스를 시작하려면

다음 예제에서는 스택의 24/7 인스턴스를 모두 시작합니다. 특정 인스턴스를 시작하려면 를 사용합니다`start-instance`.

```
aws opsworks --region us-east-1 start-stack --stack-id 8c428b08-a1a1-46ce-a5f8-feddc43771b8
```

출력 : 없음.

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 인스턴스 시작을 참조하세요.

- 자세한 API 내용은 명령 참조StartStack의 섹션을 참조하세요. AWS CLI

### `stop-instance`

다음 코드 예시에서는 `stop-instance`을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스를 중지하려면

다음 예제에서는 지정된 인스턴스를 중지합니다. 지정된 인스턴스는 인스턴스 ID로 식별됩니다. AWS OpsWorks 콘솔에서 인스턴스의 세부 정보 페이지로 이동하거나 `describe-instances` 명령을 실행하여 인스턴스 ID를 얻을 수 있습니다.

```
aws opsworks stop-instance --region us-east-1 --instance-id 3a21cfac-4a1f-4ce2-a921-b2cfba6f7771
```

를 호출하여 중지된 인스턴스를 다시 시작`start-instance`하거나 를 호출하여 인스턴스를 삭제하면 됩니다`delete-instance`.

출력 : 없음.

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 인스턴스 중지를 참조하세요.

- 자세한 API 내용은 명령 참조StopInstance의 섹션을 참조하세요. AWS CLI

### `stop-stack`

다음 코드 예시에서는 `stop-stack`을 사용하는 방법을 보여 줍니다.

AWS CLI

스택의 인스턴스를 중지하려면

다음 예제에서는 스택의 24/7 인스턴스를 모두 중지합니다. 특정 인스턴스를 중지하려면 를 사용합니다stop-instance.

```
aws opsworks --region us-east-1 stop-stack --stack-id 8c428b08-a1a1-46ce-a5f8-feddc43771b8
```

출력 : 출력이 없습니다.

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 인스턴스 중지를 참조하세요.

- 자세한 API 내용은 명령 참조StopStack의 섹션을 참조하세요. AWS CLI

### unassign-instance

다음 코드 예시에서는 unassign-instance을 사용하는 방법을 보여 줍니다.

AWS CLI

계층에서 등록된 인스턴스를 할당 취소하려면

다음 unassign-instance 명령은 연결된 계층에서 인스턴스를 할당 취소합니다.

```
aws opsworks --region us-east-1 unassign-instance --instance-id 4d6d1710-ded9-42a1-b08e-b043ad7af1e2
```

출력 : 없음.

자세한 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 등록된 인스턴스 할당 취소를 참조하세요.

- 자세한 API 내용은 명령 참조UnassignInstance의 섹션을 참조하세요. AWS CLI

### unassign-volume

다음 코드 예시에서는 unassign-volume을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스에서 볼륨 할당을 취소하려면

다음 예제에서는 인스턴스에서 등록된 Amazon Elastic Block Store(Amazon EBS) 볼륨을 할당 취소합니다. 볼륨은 볼륨 ID로 식별됩니다. 이 ID는 Amazon Elastic Compute Cloud(AmazonEC2) 볼륨 ID가 아닌 스택에 볼륨을 등록할 때 AWS OpsWorks 할당GUID하는 입니다.

```
aws opsworks --region us-east-1 unassign-volume --volume-id 8430177d-52b7-4948-9c62-e195af4703df
```

출력 : 없음.

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 Amazon EBS 볼륨 할당 취소를 참조하세요.

- 자세한 API 내용은 명령 참조UnassignVolume의 섹션을 참조하세요. AWS CLI

### update-app

다음 코드 예시에서는 update-app을 사용하는 방법을 보여 줍니다.

AWS CLI

앱을 업데이트하려면

다음 예제에서는 지정된 앱을 업데이트하여 이름을 변경합니다.

```
aws opsworks --region us-east-1 update-app --app-id 26a61ead-d201-47e3-b55c-2a7c666942f8 --name NewAppName
```

출력 : 없음.

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 앱 편집을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateApp의 섹션을 참조하세요. AWS CLI

### update-elastic-ip

다음 코드 예시에서는 update-elastic-ip을 사용하는 방법을 보여 줍니다.

AWS CLI

탄력적 IP 주소 이름을 업데이트하려면

다음 예제에서는 지정된 탄력적 IP 주소의 이름을 업데이트합니다.

```
aws opsworks --region us-east-1 update-elastic-ip --elastic-ip 54.148.130.96 --
name NewIPName
```

출력 : 없음.

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 리소스 관리를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateElasticIp의 섹션을 참조하세요. AWS CLI

### update-instance

다음 코드 예시에서는 update-instance을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스를 업데이트하려면

다음 예제에서는 지정된 인스턴스 유형을 업데이트합니다.

```
aws opsworks --region us-east-1 update-instance --instance-
id dfe18b02-5327-493d-91a4-c5c0c448927f --instance-type c3.xlarge
```

출력 : 없음.

자세한 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 인스턴스 구성 편집을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateInstance의 섹션을 참조하세요. AWS CLI

### update-layer

다음 코드 예시에서는 update-layer을 사용하는 방법을 보여 줍니다.

AWS CLI

계층을 업데이트하려면

다음 예제에서는 Amazon EBS최적화 인스턴스를 사용하도록 지정된 계층을 업데이트합니다.

```
aws opsworks --region us-east-1 update-layer --layer-
id 888c5645-09a5-4d0e-95a8-812ef1db76a4 --use-ebs-optimized-instances
```

출력 : 없음.

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 OpsWorks 계층 구성 편집을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateLayer의 섹션을 참조하세요. AWS CLI

### update-my-user-profile

다음 코드 예시에서는 update-my-user-profile을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 프로필을 업데이트하려면

다음 예제에서는 지정된 SSH 퍼블릭 키를 사용하도록 development 사용자 프로파일을 업데이트합니다. 사용자의 AWS 자격 증명은 credentials 파일(~\.aws\credentials)의 development 프로필로 표시되며 키는 작업 디렉터리의 .pem 파일에 있습니다.

```
aws opsworks --region us-east-1 --profile development update-my-user-profile --ssh-
public-key file://development_key.pem
```

출력 : 없음.

추가 정보

자세한 내용은 AWS OpsWorks 사용 AWS OpsWorks 설명서의 사용자 설정 편집을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateMyUserProfile의 섹션을 참조하세요. AWS CLI

### update-rds-db-instance

다음 코드 예시에서는 update-rds-db-instance을 사용하는 방법을 보여 줍니다.

AWS CLI

등록된 Amazon RDS DB 인스턴스를 업데이트하려면

다음 예제에서는 Amazon RDS 인스턴스의 마스터 암호 값을 업데이트합니다. 이 명령은 RDS 인스턴스의 마스터 암호를 변경하지 않고 에 제공하는 암호만 변경합니다 AWS OpsWorks. 이 암호가 RDS 인스턴스의 암호와 일치하지 않으면 애플리케이션이 데이터베이스에 연결할 수 없습니다.

```
aws opsworks --region us-east-1 update-rds-db-instance --db-password 123456789
```

출력 : 없음.

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 스택으로 Amazon RDS 인스턴스 등록을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateRdsDbInstance의 섹션을 참조하세요. AWS CLI

### `update-volume`

다음 코드 예시에서는 `update-volume`을 사용하는 방법을 보여 줍니다.

AWS CLI

등록된 볼륨을 업데이트하려면

다음 예제는 등록된 Amazon Elastic Block Store(Amazon EBS) 볼륨의 마운트 포인트를 업데이트합니다. 볼륨은 Amazon Elastic Compute Cloud(Amazon EC2) 볼륨 ID가 아닌 스택에 등록할 때 볼륨에 AWS OpsWorks 할당GUID되는 볼륨 ID로 식별됩니다.

```
aws opsworks --region us-east-1 update-volume --volume-id 8430177d-52b7-4948-9c62-
e195af4703df --mount-point /mnt/myvol
```

출력 : 없음.

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 인스턴스에 Amazon EBS 볼륨 할당을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateVolume의 섹션을 참조하세요. AWS CLI

# AWS OpsWorks CM 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS OpsWorks CM.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `associate-node`

다음 코드 예시에서는 `associate-node`을 사용하는 방법을 보여 줍니다.

AWS CLI

노드를 연결하려면

다음 `associate-node` 명령은 라는 Chef Automate 서버`i-44de882p`와 라는 노드를 연결합니다. `automate-06`즉, `automate-06` 서버가 노드를 관리하고 associate-node 명령에 의해 노드에 설치된 `chef-client` 에이전트 소프트웨어를 통해 노드에 레시피 명령을 전달합니다. 유효한 노드 이름은 EC2 인스턴스 입니다IDs.:

```
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'"
```

명령에서 반환되는 출력은 다음과 유사합니다. 출력:

```
{
 "NodeAssociationStatusToken": "AHUY8wFe4pdXtZC5DiJa5SOLp5o14DH//
rHRqHDWXxwVoNBxcEy4V7R0NOFymh7E/1HumOBPsemPQFE6dcGaiFk"
```

```
}
```

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 Chef Automate AWS OpsWorks 용 에서 노드 자동 추가를 참조하세요.

- 자세한 API 내용은 명령 참조AssociateNode의 섹션을 참조하세요. AWS CLI

### `create-backup`

다음 코드 예시에서는 `create-backup`을 사용하는 방법을 보여 줍니다.

AWS CLI

백업을 생성하려면

다음 `create-backup` 명령은 `us-east-1` 리전`automate-06`에 이름이 지정된 Chef Automate 서버의 수동 백업을 시작합니다. 명령은 `--description` 파라미터의 백업에 설명 메시지를 추가합니다.

```
aws opsworks-cm create-backup \
    --server-name 'automate-06' \
    --description "state of my infrastructure at launch"
```

출력에는 새 백업에 대한 다음과 유사한 정보가 표시됩니다.

출력:

```
{
    "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"
        }
    ],
}
```

자세한 내용은 AWS OpsWorks 사용 설명서의 Chef Automate Server AWS OpsWorks 용 백업 및 복원을 참조하세요.

- 자세한 API 내용은 명령 참조CreateBackup의 섹션을 참조하세요. AWS CLI

### `create-server`

다음 코드 예시에서는 `create-server`을 사용하는 방법을 보여 줍니다.

AWS CLI

서버를 생성하려면

다음 `create-server` 예제에서는 기본 리전`automate-06`에 이름이 지정된 새 Chef Automate 서버를 생성합니다. 기본값은 보존할 백업 수, 유지 관리 및 백업 시작 시간과 같은 대부분의 다른 설정에 사용됩니다. `create-server` 명령을 실행하기 전에 Chef Automate AWS OpsWorks 용 Opsworks 사용 설명서의 Chef Automate용 시작하기에서 사전 조건을 완료합니다. AWS

```
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"
```

출력은 새 서버에 대한 다음과 유사한 정보를 보여줍니다.

```
{
    "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" ]
    }
}
```

자세한 내용은 Chef Automate 참조용 UpdateServer의 섹션을 참조하세요. AWS OpsWorks API

- 자세한 API 내용은 명령 참조CreateServer의 섹션을 참조하세요. AWS CLI

### delete-backup

다음 코드 예시에서는 delete-backup을 사용하는 방법을 보여 줍니다.

AWS CLI

백업을 삭제하려면

다음 delete-backup 명령은 백업 ID로 식별되는 Chef Automate 서버의 수동 또는 자동 백업을 삭제합니다. 이 명령은 저장할 수 있는 최대 백업 수에 도달하거나 Amazon S3 스토리지 비용을 최소화하려는 경우에 유용합니다.

```
aws opsworks-cm delete-backup --backup-id "automate-06-2016-11-19T23:42:40.240Z"
```

출력은 백업 삭제 성공 여부를 보여줍니다.

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 Chef Automate Server AWS OpsWorks 용 백업 및 복원을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteBackup의 섹션을 참조하세요. AWS CLI

### delete-server

다음 코드 예시에서는 delete-server을 사용하는 방법을 보여 줍니다.

AWS CLI

서버를 삭제하려면

다음 delete-server 명령은 서버 이름으로 식별되는 Chef Automate 서버를 삭제합니다. 서버가 삭제된 후에는 DescribeServer 요청에 의해 더 이상 반환되지 않습니다.:

```
aws opsworks-cm delete-server --server-name "automate-06"
```

출력은 서버 삭제 성공 여부를 보여줍니다.

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 Chef Automate 서버 AWS OpsWorks 용 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteServer의 섹션을 참조하세요. AWS CLI

### `describe-account-attributes`

다음 코드 예시에서는 `describe-account-attributes`을 사용하는 방법을 보여 줍니다.

AWS CLI

계정 속성을 설명하려면

다음 `describe-account-attributes` 명령은 Chef Automate 리소스에 AWS OpsWorks 대한 계정의 사용에 대한 정보를 반환합니다.

```
aws opsworks-cm describe-account-attributes
```

명령에서 반환된 각 계정 속성 항목의 출력은 다음과 유사합니다. 출력:

```
{
 "Attributes": [
    {
       "Maximum": 5,
       "Name": "ServerLimit",
       "Used": 2
    }
 ]
}
```

추가 정보

자세한 내용은 Chef Automate 참조 DescribeAccountAttributes 의 섹션을 참조하세요. AWS OpsWorks API

- 자세한 API 내용은 명령 참조DescribeAccountAttributes의 섹션을 참조하세요. AWS CLI

### `describe-backups`

다음 코드 예시에서는 `describe-backups`을 사용하는 방법을 보여 줍니다.

AWS CLI

백업을 설명하려면

다음 describe-backups 명령은 기본 리전의 계정과 연결된 모든 백업에 대한 정보를 반환합니다.

```
aws opsworks-cm describe-backups
```

명령에서 반환되는 각 백업 항목의 출력은 다음과 유사합니다.

출력:

```
{
    "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"
        }
    ],
```

```
}
```

자세한 내용은 AWS OpsWorks 사용 설명서의 AWS OpsWorks Chef Automate 서버용 백업 및 복원을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeBackups의 섹션을 참조하세요. AWS CLI

### `describe-events`

다음 코드 예시에서는 `describe-events`을 사용하는 방법을 보여 줍니다.

AWS CLI

이벤트를 설명하려면

다음 `describe-events` 예제에서는 지정된 Chef Automate 서버와 연결된 모든 이벤트에 대한 정보를 반환합니다.

```
aws opsworks-cm describe-events \
    --server-name 'automate-06'
```

명령에서 반환된 각 이벤트 항목의 출력은 다음 예제와 유사합니다.

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS OpsWorks 사용 설명서의 일반 문제 해결 팁을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeEvents의 섹션을 참조하세요. AWS CLI

### `describe-node-association-status`

다음 코드 예시에서는 `describe-node-association-status`을 사용하는 방법을 보여 줍니다.

AWS CLI

노드 연결 상태를 설명하려면

다음 `describe-node-association-status` 명령은 노드를 라는 Chef Automate 서버와 연결하라는 요청의 상태를 반환합니다`automate-06`.

```
aws opsworks-cm describe-node-association-status --server-
name "automate-06" --node-association-status-token "AflJKl+/
GoKLZJBdDQEx0O65CDi57blQe9nKM8joSok0pQ9xr8DqApBN9/1O6sLdSvlfDEKkEx+eoCHvjoWHaOs="
```

명령에서 반환된 각 계정 속성 항목의 출력은 다음과 유사합니다. 출력:

```
{
 "NodeAssociationStatus": "IN_PROGRESS"
}
```

추가 정보

자세한 내용은 Chef Automate 참조용 DescribeNodeAssociationStatus 의 섹션을 참조하세요. AWS OpsWorks API

- 자세한 API 내용은 명령 참조DescribeNodeAssociationStatus의 섹션을 참조하세요. AWS CLI

## describe-servers

다음 코드 예시에서는 `describe-servers`을 사용하는 방법을 보여 줍니다.

AWS CLI

서버를 설명하려면

다음 `describe-servers` 명령은 계정과 연결된 모든 서버와 기본 리전에 대한 정보를 반환합니다.

```
aws opsworks-cm describe-servers
```

명령에서 반환되는 각 서버 항목의 출력은 다음과 유사합니다. 출력:

```
{
 "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" ]
    }
 ]
}
```

추가 정보

자세한 내용은 Chef Automate 가이드 DescribeServers 의 섹션을 참조하세요. AWS OpsWorks API

- 자세한 API 내용은 명령 참조DescribeServers의 섹션을 참조하세요. AWS CLI

### `disassociate-node`

다음 코드 예시에서는 `disassociate-node`을 사용하는 방법을 보여 줍니다.

AWS CLI

노드 연결을 해제하려면

다음 disassociate-node 명령은 라는 노드의 연결을 해제하여 라는 Chef Automate 서버의 관리에서 노드를 i-44de882p제거합니다automate-06. 유효한 노드 이름은 EC2 인스턴스 입니다 IDs.:

```
aws opsworks-cm disassociate-node --server-name "automate-06" --node-
name "i-43de882p" --engine-attributes "Name=CHEF_ORGANIZATION,Value='MyOrganization'
 Name=CHEF_NODE_PUBLIC_KEY,Value='Public_key_contents'"
```

명령에서 반환되는 출력은 다음과 유사합니다. 출력:

```
{
 "NodeAssociationStatusToken": "AHUY8wFe4pdXtZC5DiJa5SOLp5o14DH//
rHRqHDWXxwVoNBxcEy4V7R0NOFymh7E/1HumOBPsemPQFE6dcGaiFk"
}
```

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 AWS OpsWorks Chef Automate 서버 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DisassociateNode의 섹션을 참조하세요. AWS CLI

## restore-server

다음 코드 예시에서는 restore-server을 사용하는 방법을 보여 줍니다.

AWS CLI

서버를 복원하려면

다음 restore-server 명령은 ID가 인 백업에서 기본 리전automate-06에 이름이 지정된 Chef Automate 서버의 인플레이스 복원을 수행합니다automate-06-2016-11-22T16:13:27.998Z. 서버를 복원하면 지정된 백업이 수행된 시점에 Chef Automate 서버가 관리하고 있던 노드에 대한 연결이 복원됩니다.

aws opsworks-cm restore-server --backup-id “automate-06-2016-11-22T16:13:27.998Z” --server-name “automate-06”

출력은 명령 ID 전용입니다. 출력:

```
(None)
```

추가 정보

자세한 내용은 AWS OpsWorks 사용 설명서의 AWS OpsWorks Chef Automate 서버 실패 복원을 참조하세요.

- 자세한 API 내용은 명령 참조RestoreServer의 섹션을 참조하세요. AWS CLI

### `start-maintenance`

다음 코드 예시에서는 `start-maintenance`을 사용하는 방법을 보여 줍니다.

AWS CLI

유지 관리를 시작하려면

다음 `start-maintenance` 예제에서는 기본 리전에서 지정된 Chef Automate 또는 Puppet Enterprise 서버에 대한 유지 관리를 수동으로 시작합니다. 이 명령은 이전의 자동 유지 관리 시도가 실패하고 유지 관리 실패의 근본 원인이 해결된 경우에 유용합니다.

```
aws opsworks-cm start-maintenance \
    --server-name 'automate-06'
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS OpsWorks 사용 설명서의 시스템 유지 관리(Puppet Enterprise 서버) 또는 시스템 유지 관리(Chef Automate 서버)를 참조하세요.

- 자세한 API 내용은 명령 참조StartMaintenance의 섹션을 참조하세요. AWS CLI

### update-server-engine-attributes

다음 코드 예시에서는 update-server-engine-attributes을 사용하는 방법을 보여 줍니다.

AWS CLI

서버 엔진 속성을 업데이트하려면

다음 update-server-engine-attributes 명령은 라는 Chef Automate 서버의 CHEF_PIVOTAL_KEY 엔진 속성 값을 업데이트합니다automate-06. 현재 다른 엔진 속성의 값을 변경할 수 없습니다.

```
aws opsworks-cm update-server-engine-attributes \
    --attribute-name CHEF_PIVOTAL_KEY \
    --attribute-value "new key value" \
    --server-name "automate-06"
```

출력은 업데이트된 서버에 대한 다음과 유사한 정보를 보여줍니다.

```
{
    "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" ]
    }
}
```

자세한 내용은 Chef Automate 참조용 UpdateServerEngineAttributes의 섹션을 참조하세요. AWS OpsWorks API

- 자세한 API 내용은 명령 참조UpdateServerEngineAttributes의 섹션을 참조하세요. AWS CLI

### update-server

다음 코드 예시에서는 update-server을 사용하는 방법을 보여 줍니다.

AWS CLI

서버를 업데이트하려면

다음 update-server 명령은 기본 리전에서 지정된 Chef Automate 서버의 유지 관리 시작 시간을 업데이트합니다. --preferred-maintenance-window 파라미터가 추가되어 서버 유지 관리의 시작 날짜 및 시간을 월요일 오전 9시 15분으로 변경합니다. UTC.:

```
aws opsworks-cm update-server \
    --server-name "automate-06" \
    --preferred-maintenance-window "Mon:09:15"
```

출력은 업데이트된 서버에 대한 다음과 유사한 정보를 보여줍니다.

```
{
    "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" ]
    }
}
```

자세한 내용은 Chef Automate 참조 UpdateServer의 섹션을 참조하세요. AWS OpsWorks API

- 자세한 API 내용은 명령 참조UpdateServer의 섹션을 참조하세요. AWS CLI

## 를 사용한 조직 예제 AWS CLI

다음 코드 예제에서는 조직과 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `accept-handshake`

다음 코드 예시에서는 `accept-handshake`을 사용하는 방법을 보여 줍니다.

AWS CLI

다른 계정에서 핸드셰이크를 수락하려면

조직의 소유자인 Bill은 이전에 Juan의 계정을 초대하여 조직에 가입했습니다. 다음 예제는 핸드셰이크를 수락하여 초대에 동의하는 Juan의 계정을 보여줍니다.

```
aws organizations accept-handshake --handshake-id h-examplehandshakeid111
```

출력은 다음과 같이 표시됩니다.

```
{
        "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"
```

```
        }
}
```

- 자세한 API 내용은 명령 참조AcceptHandshake의 섹션을 참조하세요. AWS CLI

### attach-policy

다음 코드 예시에서는 attach-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

정책을 루트, OU 또는 계정에 연결하는 방법

예 1

다음 예제에서는 서비스 제어 정책(SCP)을 OU에 연결하는 방법을 보여줍니다.

```
aws organizations attach-policy
                --policy-id p-examplepolicyid111
                --target-id ou-examplerootid111-exampleouid111
```

예제 2

다음 예시에서는 계정에 서비스 제어 정책을 직접 연결하는 방법을 보여줍니다.

```
aws organizations attach-policy
                --policy-id p-examplepolicyid111
                --target-id 333333333333
```

- 자세한 API 내용은 명령 참조AttachPolicy의 섹션을 참조하세요. AWS CLI

### cancel-handshake

다음 코드 예시에서는 cancel-handshake을 사용하는 방법을 보여 줍니다.

AWS CLI

다른 계정에서 전송된 핸드셰이크를 취소하려면

Bill은 이전에 Susan의 계정에 조직 가입 초대장을 보냈습니다. 그는 마음을 바꾸어 Susan이 초대를 수락하기 전에 이를 취소하기로 결정합니다. 다음 예제에서는 Bill의 취소를 보여줍니다.

```
aws organizations cancel-handshake --handshake-id h-examplehandshakeid111
```

출력에는 상태가 현재 임을 보여주는 핸드셰이크 객체가 포함됩니다CANCELED.

```
{
        "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
        }
}
```

- 자세한 API 내용은 명령 참조CancelHandshake의 섹션을 참조하세요. AWS CLI

### `create-account`

다음 코드 예시에서는 `create-account`을 사용하는 방법을 보여 줍니다.

AWS CLI

자동으로 조직의 일부가 되는 멤버 계정을 생성하는 방법

다음 예시에서는 조직 내에 멤버 계정을 생성하는 방법을 보여줍니다. 멤버 계정은 Production Account라는 이름과 susan@example.com이라는 이메일 주소로 구성됩니다. roleName 파라미터가 지정되지 OrganizationAccountAccessRole 않았기 때문에 조직은 기본 이름인 를 사용하여 IAM 역할을 자동으로 생성합니다. 또한 충분한 권한을 가진 IAM 사용자 또는 역할이 계정 결제 데이터에 액세스할 수 있도록 허용하는 설정은 IamUserAccessToBilling 파라미터가 지정되지 ALLOW 않았기 때문에 의 기본값으로 설정됩니다. 조직은 Susan에게 “환영합니다 AWS”라는 이메일을 자동으로 보냅니다.

```
aws organizations create-account --email susan@example.com --account-
name "Production Account"
```

출력에는 현재 `IN_PROGRESS` 상태임을 보여주는 요청 객체가 포함됩니다.

```
{
        "CreateAccountStatus": {
                "State": "IN_PROGRESS",
                "Id": "car-examplecreateaccountrequestid111"
        }
}
```

나중에 명령에 Id 응답 값을 파라미터 값으로 제공하여 요청의 현재 상태를 쿼리할 describe-create-account-status 수 있습니다 create-account-request-id.

자세한 내용은 AWS Organizations 사용 설명서의 조직에서 AWS 계정 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateAccount의 섹션을 참조하세요. AWS CLI

### `create-organization`

다음 코드 예시에서는 `create-organization`을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: 새 조직을 생성하는 방법

Bill은 111111111111 계정의 보안 인증 정보를 사용하여 조직을 만들려고 합니다. 다음 예시에서는 해당 계정이 새 조직의 마스터 계정이 되는 것을 보여줍니다. Bill이 기능 세트를 지정하지 않기 때문에 새 조직에서는 기본적으로 모든 기능이 활성화되고 서비스 제어 정책은 루트에서 활성화됩니다.

```
aws organizations create-organization
```

출력에는 새 조직에 대한 세부 정보가 포함된 조직 객체가 포함됩니다.

```
{
        "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"
        }
}
```

예 2: 통합 결제 기능만 활성화된 새 조직을 생성하는 방법

다음 예시에서는 통합 결제 기능만 지원하는 조직을 만듭니다.

```
aws organizations create-organization --feature-set CONSOLIDATED_BILLING
```

출력에는 새 조직에 대한 세부 정보가 포함된 조직 객체가 포함됩니다.

```
{
        "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"
        }
}
```

자세한 내용은 AWS Organizations 사용 설명서의 조직 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateOrganization의 섹션을 참조하세요. AWS CLI

### create-organizational-unit

다음 코드 예시에서는 create-organizational-unit을 사용하는 방법을 보여 줍니다.

AWS CLI

루트 또는 상위 OU에 OU를 생성하는 방법

다음 예시에서는 AccountingOU라는 OU를 생성하는 방법을 보여줍니다.

```
aws organizations create-organizational-unit --parent-id r-examplerootid111 --
name AccountingOU
```

출력에는 새 OU에 대한 세부 정보가 포함된 organizationalUnit 객체가 포함됩니다.

```
{
```

```
        "OrganizationalUnit": {
                "Id": "ou-examplerootid111-exampleouid111",
                "Arn": "arn:aws:organizations::111111111111:ou/o-exampleorgid/ou-
examplerootid111-exampleouid111",
                "Name": "AccountingOU"
        }
}
```

- 자세한 API 내용은 명령 참조CreateOrganizationalUnit의 섹션을 참조하세요. AWS CLI

### `create-policy`

다음 코드 예시에서는 `create-policy`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 정책의 텍스트 소스 파일로 JSON 정책을 생성하려면

다음 예제에서는 라는 서비스 제어 정책(SCP)을 생성하는 방법을 보여줍니다AllowAllS3Actions. 정책 콘텐츠는 policy.json이라는 로컬 컴퓨터에 있는 파일에서 가져온 것입니다.

```
aws organizations create-policy --content file://policy.json --
name AllowAllS3Actions, --type SERVICE_CONTROL_POLICY --description "Allows
 delegation of all S3 actions"
```

출력에는 새 정책에 대한 세부 정보가 포함된 정책 객체가 포함됩니다.

```
{
        "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"
                }
        }
}
```

예제 2: 정책을 파라미터로 사용하여 JSON 정책을 생성하려면

다음 예제에서는 파라미터에 정책 내용을 JSON 문자열로 포함SCP시켜 동일한 를 생성하는 방법을 보여줍니다. 파라미터에서 문자열을 큰 따옴표로 묶은 리터럴로 취급하도록 하려면 큰따옴표 앞의 백슬래시로 문자열을 이스케이프 처리해야 합니다.

```
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"
```

조직에서 정책을 만들고 사용하는 방법에 대한 자세한 내용은 AWS Organizations 사용 설명서의 조직 정책 관리를 참조하세요.

- 자세한 API 내용은 명령 참조CreatePolicy의 섹션을 참조하세요. AWS CLI

## decline-handshake

다음 코드 예시에서는 decline-handshake을 사용하는 방법을 보여 줍니다.

AWS CLI

다른 계정에서 전송된 핸드셰이크를 거부하려면

다음 예에서는 계정 222222222222의 소유자인 관리자인 Susan이 Bill의 조직에 가입하라는 초대를 거부하는 것을 보여줍니다. DeclineHandshake 작업은 핸드셰이크 객체를 반환하여 상태가 현재임을 보여줍니다DECLINED.

```
aws organizations decline-handshake --handshake-id h-examplehandshakeid111
```

출력에는 의 새 상태를 보여주는 핸드셰이크 객체가 포함됩니다DECLINED.

```
{
        "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"
        }
}
```

- 자세한 API 내용은 명령 참조DeclineHandshake의 섹션을 참조하세요. AWS CLI

### `delete-organization`

다음 코드 예시에서는 `delete-organization`을 사용하는 방법을 보여 줍니다.

AWS CLI

조직을 삭제하는 방법

다음 예시에서는 조직을 삭제하는 방법을 보여줍니다. 이 작업을 수행하려면 조직의 마스터 계정 관리자여야 합니다. 이 예제에서는 이전에 조직에서 모든 멤버 계정, OUs및 정책을 제거했다고 가정합니다.

```
aws organizations delete-organization
```

- 자세한 API 내용은 명령 참조DeleteOrganization의 섹션을 참조하세요. AWS CLI

### `delete-organizational-unit`

다음 코드 예시에서는 `delete-organizational-unit`을 사용하는 방법을 보여 줍니다.

AWS CLI

OU를 삭제하는 방법

다음 예제에서는 OU를 삭제하는 방법을 보여줍니다. 이 예제에서는 이전에 OUOUs에서 모든 계정 및 기타를 제거했다고 가정합니다.

```
aws organizations delete-organizational-unit --organizational-unit-id ou-
examplerootid111-exampleouid111
```

- 자세한 API 내용은 명령 참조DeleteOrganizationalUnit의 섹션을 참조하세요. AWS CLI

### `delete-policy`

다음 코드 예시에서는 `delete-policy`을 사용하는 방법을 보여 줍니다.

AWS CLI

정책을 삭제하는 방법

다음 예시에서는 조직에서 정책을 삭제하는 방법을 보여줍니다. 이 예시에서는 이전에 정책을 모든 엔터티에서 분리했다고 가정합니다.

```
aws organizations delete-policy --policy-id p-examplepolicyid111
```

- 자세한 API 내용은 명령 참조DeletePolicy의 섹션을 참조하세요. AWS CLI

### `describe-account`

다음 코드 예시에서는 `describe-account`을 사용하는 방법을 보여 줍니다.

AWS CLI

계정에 대한 세부 정보를 가져오려면

다음 예제에서는 계정에 대한 세부 정보를 요청하는 방법을 보여줍니다.

```
aws organizations describe-account --account-id 555555555555
```

출력은 계정에 대한 세부 정보가 포함된 계정 객체를 보여줍니다.

```
{
        "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"
        }
}
```

- 자세한 API 내용은 명령 참조DescribeAccount의 섹션을 참조하세요. AWS CLI

### `describe-create-account-status`

다음 코드 예시에서는 `describe-create-account-status`을 사용하는 방법을 보여 줍니다.

AWS CLI

계정 생성 요청에 대한 최신 상태를 확인하려면

다음 예제는 조직에서 계정을 생성하기 위해 이전 요청의 최신 상태를 요청하는 방법을 보여줍니다. 지정된 --request-id는 계정 생성에 대한 원래 호출의 응답에서 가져옵니다. 계정 생성 요청은 Organizations가 계정 생성을 성공적으로 완료했음을 상태 필드에 표시합니다.

명령:

```
aws organizations describe-create-account-status --create-account-request-id car-examplecreateaccountrequestid111
```

출력:

```
{
  "CreateAccountStatus": {
    "State": "SUCCEEDED",
    "AccountId": "555555555555",
    "AccountName": "Beta account",
    "RequestedTimestamp": 1470684478.687,
    "CompletedTimestamp": 1470684532.472,
    "Id": "car-examplecreateaccountrequestid111"
  }
}
```

- 자세한 API 내용은 명령 참조DescribeCreateAccountStatus의 섹션을 참조하세요. AWS CLI

### describe-handshake

다음 코드 예시에서는 describe-handshake을 사용하는 방법을 보여 줍니다.

AWS CLI

핸드셰이크에 대한 정보를 가져오려면

다음 예제에서는 핸드셰이크에 대한 세부 정보를 요청하는 방법을 보여줍니다. 핸드셰이크 ID는 원래 호출에서 로InviteAccountToOrganization, 또는 호출에서 ListHandshakesForAccount 또는 로 가져옵니다ListHandshakesForOrganization.

```
aws organizations describe-handshake --handshake-id h-examplehandshakeid111
```

출력에는 요청된 핸드셰이크에 대한 모든 세부 정보가 포함된 핸드셰이크 객체가 포함됩니다.

```
{
        "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"
        }
}
```

- 자세한 API 내용은 명령 참조DescribeHandshake의 섹션을 참조하세요. AWS CLI

### `describe-organization`

다음 코드 예시에서는 `describe-organization`을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 조직에 대한 정보를 가져오려면

다음 예제에서는 조직에 대한 세부 정보를 요청하는 방법을 보여줍니다.

```
aws organizations describe-organization
```

출력에는 조직에 대한 세부 정보가 있는 조직 객체가 포함됩니다.

```
{
        "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"
                        }
                ]
        }
}
```

- 자세한 API 내용은 명령 참조DescribeOrganization의 섹션을 참조하세요. AWS CLI

## describe-organizational-unit

다음 코드 예시에서는 describe-organizational-unit을 사용하는 방법을 보여 줍니다.

AWS CLI

OU에 대한 정보를 가져오려면

다음 describe-organizational-unit 예제에서는 OU에 대한 세부 정보를 요청합니다.

```
aws organizations describe-organizational-unit \
```

```
    --organizational-unit-id ou-examplerootid111-exampleouid111
```

출력:

```
{
    "OrganizationalUnit": {
        "Name": "Accounting Group",
        "Arn": "arn:aws:organizations::123456789012:ou/o-exampleorgid/ou-
examplerootid111-exampleouid111",
        "Id": "ou-examplerootid111-exampleouid111"
    }
}
```

- 자세한 API 내용은 명령 참조DescribeOrganizationalUnit의 섹션을 참조하세요. AWS CLI

### describe-policy

다음 코드 예시에서는 describe-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

정책에 대한 정보를 가져오는 방법

다음 예시에서는 정책에 대한 정보를 요청하는 방법을 보여줍니다.

```
aws organizations describe-policy --policy-id p-examplepolicyid111
```

출력에는 정책에 대한 세부 정보가 포함된 정책 객체가 포함됩니다.

```
{
        "Policy": {
                "Content": "{\n  \"Version\": \"2012-10-17\",\n  \"Statement\": [\n
    {\n      \"Effect\": \"Allow\",\n      \"Action\": \"*\",\n      \"Resource\":
 \"*\"\n    }\n  ]\n}",
                "PolicySummary": {
                        "Arn": "arn:aws:organizations::111111111111:policy/o-
exampleorgid/service_control_policy/p-examplepolicyid111",
                        "Type": "SERVICE_CONTROL_POLICY",
                        "Id": "p-examplepolicyid111",
                        "AwsManaged": false,
                        "Name": "AllowAllS3Actions",
                        "Description": "Enables admins to delegate S3 permissions"
```

```
                }
        }
}
```

- 자세한 API 내용은 명령 참조DescribePolicy의 섹션을 참조하세요. AWS CLI

### `detach-policy`

다음 코드 예시에서는 `detach-policy`을 사용하는 방법을 보여 줍니다.

AWS CLI

정책을 루트, OU 또는 계정에서 분리하는 방법

다음 예시에서는 OU에서 정책을 분리하는 방법을 보여줍니다.

```
aws organizations  detach-policy  --target-id ou-examplerootid111-exampleouid111 --
policy-id p-examplepolicyid111
```

- 자세한 API 내용은 명령 참조DetachPolicy의 섹션을 참조하세요. AWS CLI

### `disable-policy-type`

다음 코드 예시에서는 `disable-policy-type`을 사용하는 방법을 보여 줍니다.

AWS CLI

루트에서 정책 유형을 비활성화하려면

다음 예제에서는 루트에서 서비스 제어 정책(SCP) 정책 유형을 비활성화하는 방법을 보여줍니다.

```
aws organizations disable-policy-type --root-id r-examplerootid111 --policy-
type SERVICE_CONTROL_POLICY
```

출력은 PolicyTypes 응답 요소에 더 이상 SERVICE_CONTROL_가 포함되지 않음을 보여줍니다 POLICY.

```
{
        "Root": {
                "PolicyTypes": [],
                "Name": "Root",
```

```
                "Id": "r-examplerootid111",
                "Arn": "arn:aws:organizations::111111111111:root/o-exampleorgid/r-
examplerootid111"
        }
}
```

- 자세한 API 내용은 명령 참조DisablePolicyType의 섹션을 참조하세요. AWS CLI

### `enable-all-features`

다음 코드 예시에서는 `enable-all-features`을 사용하는 방법을 보여 줍니다.

AWS CLI

조직의 모든 기능을 활성화하려면

이 예제는 조직의 모든 초대된 계정에 조직의 활성화된 모든 기능을 승인하도록 요청하는 관리자를 보여줍니다. AWS Organizations는 모든 초대된 멤버 계정에 등록된 주소로 이메일을 보내 소유자에게 전송된 핸드셰이크를 수락하여 모든 기능에 대한 변경을 승인하도록 요청합니다. 초대된 모든 멤버 계정이 핸드셰이크를 수락한 후 조직 관리자는 모든 기능에 대한 변경을 완료할 수 있으며, 적절한 권한이 있는 계정은 정책을 생성하고 루트, OUs및 계정에 적용할 수 있습니다.

```
aws organizations enable-all-features
```

출력은 승인을 위해 초대된 모든 멤버 계정으로 전송되는 핸드셰이크 객체입니다.

```
{
        "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"
        }
}
```

- 자세한 API 내용은 명령 참조EnableAllFeatures의 섹션을 참조하세요. AWS CLI

### enable-policy-type

다음 코드 예시에서는 enable-policy-type을 사용하는 방법을 보여 줍니다.

AWS CLI

루트에서 정책 유형 사용을 활성화하려면

다음 예제에서는 루트에서 서비스 제어 정책(SCP) 정책 유형을 활성화하는 방법을 보여줍니다.

```
aws organizations enable-policy-type --root-id r-examplerootid111 --policy-
type SERVICE_CONTROL_POLICY
```

출력은 이제 활성화SCPs되었음을 나타내는 policyTypes 응답 요소가 있는 루트 객체를 보여줍니다.

```
{
        "Root": {
                "PolicyTypes": [
                        {
                                "Status":"ENABLED",
                                "Type":"SERVICE_CONTROL_POLICY"
                        }
                ],
                "Id": "r-examplerootid111",
                "Name": "Root",
                "Arn": "arn:aws:organizations::111111111111:root/o-exampleorgid/r-
examplerootid111"
        }
}
```

- 자세한 API 내용은 명령 참조EnablePolicyType의 섹션을 참조하세요. AWS CLI

### `invite-account-to-organization`

다음 코드 예시에서는 `invite-account-to-organization`을 사용하는 방법을 보여 줍니다.

AWS CLI

조직에 가입하도록 계정을 초대하려면

다음 예제는 bill@example.com이 소유한 마스터 계정을 juan@example.com이 소유한 계정을 조직에 가입하도록 초대하는 것을 보여줍니다.

```
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."
```

출력에는 초대된 계정으로 전송되는 내용을 보여주는 핸드셰이크 구조가 포함됩니다.

```
{
        "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"
                }
}
```

- 자세한 API 내용은 명령 참조InviteAccountToOrganization의 섹션을 참조하세요. AWS CLI

### `leave-organization`

다음 코드 예시에서는 `leave-organization`을 사용하는 방법을 보여 줍니다.

AWS CLI

조직을 멤버 계정으로 두려면

다음 예제는 현재 멤버인 조직을 떠나도록 요청하는 멤버 계정의 관리자를 보여줍니다.

```
aws organizations leave-organization
```

- 자세한 API 내용은 명령 참조LeaveOrganization의 섹션을 참조하세요. AWS CLI

### `list-accounts-for-parent`

다음 코드 예시에서는 `list-accounts-for-parent`을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 상위 루트 또는 OU의 모든 계정 목록을 검색하려면

다음 예제에서는 OU에서 계정 목록을 요청하는 방법을 보여줍니다.

```
aws organizations list-accounts-for-parent --parent-id ou-examplerootid111-exampleouid111
```

출력에는 계정 요약 객체 목록이 포함됩니다.

```
{
        "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"
                }
        ]
}
```

- 자세한 API 내용은 명령 참조ListAccountsForParent의 섹션을 참조하세요. AWS CLI

### `list-accounts`

다음 코드 예시에서는 `list-accounts`을 사용하는 방법을 보여 줍니다.

AWS CLI

조직의 모든 계정 목록을 검색하는 방법

다음 예시에서는 조직의 계정 목록을 요청하는 방법을 보여줍니다.

```
aws organizations list-accounts
```

출력에는 계정 요약 객체 목록이 포함됩니다.

```
{
        "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"
                }
        ]
}
```

- 자세한 API 내용은 명령 참조ListAccounts의 섹션을 참조하세요. AWS CLI

### `list-children`

다음 코드 예시에서는 `list-children`을 사용하는 방법을 보여 줍니다.

AWS CLI

상위 OU 또는 루트OUs의 하위 계정 및 를 검색하려면

다음 예제에서는 해당 계정 4444444444444이 포함된 루트 또는 OU를 나열하는 방법을 보여줍니다.

```
aws organizations list-children --child-type ORGANIZATIONAL_UNIT --parent-id ou-
examplerootid111-exampleouid111
```

출력은 상위에 OUs 포함된 두 하위를 보여줍니다.

```
{
        "Children": [
                {
                        "Id": "ou-examplerootid111-exampleouid111",
                        "Type":"ORGANIZATIONAL_UNIT"
                },
                {
                        "Id":"ou-examplerootid111-exampleouid222",
                        "Type":"ORGANIZATIONAL_UNIT"
                }
        ]
}
```

- 자세한 API 내용은 명령 참조ListChildren의 섹션을 참조하세요. AWS CLI

### `list-create-account-status`

다음 코드 예시에서는 `list-create-account-status`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 현재 조직에서 수행된 계정 생성 요청 목록을 검색하려면

다음 예제에서는 성공적으로 완료된 조직에 대한 계정 생성 요청 목록을 요청하는 방법을 보여줍니다.

```
aws organizations list-create-account-status --states SUCCEEDED
```

출력에는 각 요청에 대한 정보가 포함된 객체 배열이 포함됩니다.

```
{
        "CreateAccountStatuses": [
                {
                        "AccountId": "444444444444",
                        "AccountName": "Developer Test Account",
                        "CompletedTimeStamp": 1481835812.143,
                        "Id": "car-examplecreateaccountrequestid111",
                        "RequestedTimeStamp": 1481829432.531,
                        "State": "SUCCEEDED"
                }
        ]
}
```

예제 2: 현재 조직에서 진행 중인 계정 생성 요청 목록을 검색하려면

다음 예제에서는 조직에 대해 진행 중인 계정 생성 요청 목록을 가져옵니다.

```
aws organizations list-create-account-status --states IN_PROGRESS
```

출력에는 각 요청에 대한 정보가 포함된 객체 배열이 포함됩니다.

```
{
        "CreateAccountStatuses": [
                {
                  "State": "IN_PROGRESS",
```

```
                "Id": "car-examplecreateaccountrequestid111",
                "RequestedTimeStamp": 1481829432.531,
                "AccountName": "Production Account"
            }
        ]
}
```

- 자세한 API 내용은 명령 참조ListCreateAccountStatus의 섹션을 참조하세요. AWS CLI

### `list-handshakes-for-account`

다음 코드 예시에서는 `list-handshakes-for-account`을 사용하는 방법을 보여 줍니다.

AWS CLI

계정으로 전송된 핸드셰이크 목록을 검색하려면

다음 예제에서는 작업을 호출하는 데 사용된 자격 증명의 계정과 연결된 모든 핸드셰이크 목록을 가져오는 방법을 보여줍니다.

```
aws organizations list-handshakes-for-account
```

출력에는 현재 상태를 포함하여 각 핸드셰이크에 대한 정보가 포함된 핸드셰이크 구조 목록이 포함됩니다.

```
{
        "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"
        }
}
```

- 자세한 API 내용은 명령 참조ListHandshakesForAccount의 섹션을 참조하세요. AWS CLI

### `list-handshakes-for-organization`

다음 코드 예시에서는 `list-handshakes-for-organization`을 사용하는 방법을 보여 줍니다.

AWS CLI

조직과 연결된 핸드셰이크 목록을 검색하려면

다음 예제에서는 현재 조직과 연결된 핸드셰이크 목록을 가져오는 방법을 보여줍니다.

```
aws organizations list-handshakes-for-organization
```

출력에는 두 개의 핸드셰이크가 표시됩니다. 첫 번째는 Juan의 계정에 대한 초대이며 상태가 표시됩니다OPEN. 두 번째는 Anika의 계정에 대한 초대이며 의 상태를 보여줍니다ACCEPTED.

```
{
        "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."
                                }
                        ]
                }
        ]
}
```

- 자세한 API 내용은 명령 참조ListHandshakesForOrganization의 섹션을 참조하세요. AWS CLI

### `list-organizational-units-for-parent`

다음 코드 예시에서는 `list-organizational-units-for-parent`을 사용하는 방법을 보여 줍니다.

AWS CLI

상위 OU 또는 루트OUs에서 의 목록을 검색하려면

다음 예제에서는 지정된 루트OUs에서 목록을 가져오는 방법을 보여줍니다.

```
aws organizations list-organizational-units-for-parent --parent-id r-
examplerootid111
```

출력은 지정된 루트에 두 개의 가 포함되어 OUs 있고 각 루트의 세부 정보를 보여줍니다.

```
{
        "OrganizationalUnits": [
                {
                        "Name": "AccountingDepartment",
                        "Arn": "arn:aws:organizations::o-exampleorgid:ou/r-
examplerootid111/ou-examplerootid111-exampleouid111"
                },
                {
                        "Name": "ProductionDepartment",
```

```
                        "Arn": "arn:aws:organizations::o-exampleorgid:ou/r-
examplerootid111/ou-examplerootid111-exampleouid222"
                }
        ]
}
```

- 자세한 API 내용은 명령 참조ListOrganizationalUnitsForParent의 섹션을 참조하세요. AWS CLI

### `list-parents`

다음 코드 예시에서는 `list-parents`을 사용하는 방법을 보여 줍니다.

AWS CLI

계정 OUs 또는 하위 OU의 상위 또는 루트를 나열하려면

다음 예제에서는 해당 계정 444444444444이 포함된 루트 또는 상위 OU를 나열하는 방법을 보여줍니다.

```
aws organizations list-parents --child-id 444444444444
```

출력은 지정된 계정이 지정된 ID를 가진 OU에 있음을 보여줍니다.

```
{
  "Parents": [
        {
          "Id": "ou-examplerootid111-exampleouid111",
          "Type": "ORGANIZATIONAL_UNIT"
        }
  ]
}
```

- 자세한 API 내용은 명령 참조ListParents의 섹션을 참조하세요. AWS CLI

### `list-policies-for-target`

다음 코드 예시에서는 `list-policies-for-target`을 사용하는 방법을 보여 줍니다.

AWS CLI

계정에 직접 SCPs 연결된 의 목록을 검색하려면

다음 예제에서는 필터 파라미터에 지정된 대로 계정에 직접 연결된 모든 서비스 제어 정책(SCPs) 목록을 가져오는 방법을 보여줍니다.

```
aws organizations list-policies-for-target --filter SERVICE_CONTROL_POLICY --target-
id 444444444444
```

출력에는 정책에 대한 요약 정보가 포함된 정책 구조 목록이 포함됩니다. 목록에는 OU 계층 구조의 위치에서 상속되기 때문에 계정에 적용되는 정책이 포함되지 않습니다.

```
{
        "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."
                }
        ]
}
```

- 자세한 API 내용은 명령 참조ListPoliciesForTarget의 섹션을 참조하세요. AWS CLI

### `list-policies`

다음 코드 예시에서는 `list-policies`을 사용하는 방법을 보여 줍니다.

AWS CLI

특정 유형의 조직에 있는 모든 정책 목록을 검색하는 방법

다음 예제에서는 필터 파라미터에 지정된 SCPs대로 목록을 가져오는 방법을 보여줍니다.

```
aws organizations list-policies --filter SERVICE_CONTROL_POLICY
```

출력에는 요약 정보가 포함된 정책 목록이 포함됩니다.

```
{
        "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"
                }
        ]
}
```

- 자세한 API 내용은 명령 참조ListPolicies의 섹션을 참조하세요. AWS CLI

### `list-roots`

다음 코드 예시에서는 `list-roots`을 사용하는 방법을 보여 줍니다.

AWS CLI

조직의 루트 목록을 검색하려면

이 예제에서는 조직의 루트 목록을 가져오는 방법을 보여줍니다.

```
aws organizations list-roots
```

출력에는 요약 정보가 포함된 루트 구조 목록이 포함됩니다.

```
{
        "Roots": [
                {
                        "Name": "Root",
                        "Arn": "arn:aws:organizations::111111111111:root/o-
exampleorgid/r-examplerootid111",
                        "Id": "r-examplerootid111",
                        "PolicyTypes": [
                                {
                                        "Status":"ENABLED",
                                        "Type":"SERVICE_CONTROL_POLICY"
                                }
                        ]
                }
        ]
}
```

- 자세한 API 내용은 명령 참조ListRoots의 섹션을 참조하세요. AWS CLI

### list-targets-for-policy

다음 코드 예시에서는 `list-targets-for-policy`을 사용하는 방법을 보여 줍니다.

AWS CLI

정책이 연결된 루트, OUs및 계정 목록을 검색하려면

다음 예제에서는 지정된 정책이 연결된 루트, OUs및 계정 목록을 가져오는 방법을 보여줍니다.

```
aws organizations list-targets-for-policy --policy-id p-FullAWSAccess
```

출력에는 정책이 연결된 루트, OUs및 계정에 대한 요약 정보가 포함된 연결 객체 목록이 포함됩니다.

```
{
        "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-exampleouid111",
                        "Name":"Accounting",
                        "TargetId":"ou-examplerootid111-exampleouid111",
                        "Type":"ORGANIZATIONAL_UNIT"
                }
        ]
}
```

- 자세한 API 내용은 명령 참조ListTargetsForPolicy의 섹션을 참조하세요. AWS CLI

### `move-account`

다음 코드 예시에서는 `move-account`을 사용하는 방법을 보여 줍니다.

AWS CLI

루트 또는 간에 계정을 이동하려면 OUs

다음 예제에서는 조직의 마스터 계정을 루트에서 OU로 이동하는 방법을 보여줍니다.

```
aws organizations move-account --account-id 333333333333 --source-parent-id r-
examplerootid111 --destination-parent-id ou-examplerootid111-exampleouid111
```

- 자세한 API 내용은 명령 참조MoveAccount의 섹션을 참조하세요. AWS CLI

### remove-account-from-organization

다음 코드 예시에서는 remove-account-from-organization을 사용하는 방법을 보여 줍니다.

AWS CLI

조직에서 마스터 계정으로 계정을 제거하려면

다음 예제에서는 조직에서 계정을 제거하는 방법을 보여줍니다.

```
aws organizations remove-account-from-organization --account-id 333333333333
```

- 자세한 API 내용은 명령 참조RemoveAccountFromOrganization의 섹션을 참조하세요. AWS CLI

### update-organizational-unit

다음 코드 예시에서는 update-organizational-unit을 사용하는 방법을 보여 줍니다.

AWS CLI

OU의 이름을 바꾸려면

이 예제에서는 OU의 이름을 바꾸는 방법을 보여줍니다. 이 예제에서는 OU의 이름이 “AccountingOU”로 변경됩니다.

```
aws organizations update-organizational-unit --organizational-unit-id ou-
examplerootid111-exampleouid111 --name AccountingOU
```

출력에는 새 이름이 표시됩니다.

```
{
        "OrganizationalUnit": {
                "Id": "ou-examplerootid111-exampleouid111"
                "Name": "AccountingOU",
                "Arn": "arn:aws:organizations::111111111111:ou/o-exampleorgid/ou-
examplerootid111-exampleouid111""
        }
}
```

- 자세한 API 내용은 명령 참조UpdateOrganizationalUnit의 섹션을 참조하세요. AWS CLI

### update-policy

다음 코드 예시에서는 update-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 정책 이름 바꾸기

다음 update-policy 예제에서는 정책의 이름을 바꾸고 새 설명을 제공합니다.

```
aws organizations update-policy \
    --policy-id p-examplepolicyid111 \
    --name Renamed-Policy \
    --description "This description replaces the original."
```

출력에는 새 이름과 설명이 표시됩니다.

```
{
    "Policy": {
        "Content": "{\n  \"Version\":\"2012-10-17\",\n  \"Statement\":{\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"
        }
    }
}
```

예제 2: 정책의 JSON 텍스트 콘텐츠를 바꾸려면

다음 예제에서는 SCP 이전 예제에서 의 JSON 텍스트를 대신 S3를 허용하는 새 JSON 정책 텍스트 문자열로 바꾸는 방법을 보여줍니다EC2.

```
aws organizations update-policy \
    --policy-id p-examplepolicyid111 \
```

```
    --content "{\"Version\":\"2012-10-17\",\"Statement\":{\"Effect\":\"Allow\",
\"Action\":\"s3:*\",\"Resource\":\"*\"}}"
```

출력에는 새 콘텐츠가 표시됩니다.

```
{
    "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"
        }
    }
}
```

- 자세한 API 내용은 명령 참조UpdatePolicy의 섹션을 참조하세요. AWS CLI

## AWS Outposts 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS Outposts.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

작업

### `get-outpost-instance-types`

다음 코드 예시에서는 `get-outpost-instance-types`을 사용하는 방법을 보여 줍니다.

AWS CLI

Outpost에서 인스턴스 유형을 가져오려면

다음 `get-outpost-instance-types` 예제에서는 지정된 Outpost의 인스턴스 유형을 가져옵니다.

```
aws outposts get-outpost-instance-types \
    --outpost-id op-0ab23c4567EXAMPLE
```

출력:

```
{
    "InstanceTypes": [
        {
            "InstanceType": "c5d.large"
        },
        {
            "InstanceType": "i3en.24xlarge"
        },
        {
            "InstanceType": "m5d.large"
        },
        {
            "InstanceType": "r5d.large"
        }
    ],
    "OutpostId": "op-0ab23c4567EXAMPLE",
    "OutpostArn": "arn:aws:outposts:us-west-2:123456789012:outpost/
op-0ab23c4567EXAMPLE"
}
```

자세한 내용은 Outposts 사용 설명서의 Outpost에서 인스턴스 시작을 참조하세요. AWS

- 자세한 API 내용은 명령 참조GetOutpostInstanceTypes의 섹션을 참조하세요. AWS CLI

### `get-outpost`

다음 코드 예시에서는 `get-outpost`을 사용하는 방법을 보여 줍니다.

AWS CLI

Outpost 세부 정보를 가져오려면

다음 `get-outpost` 예제에서는 지정된 Outpost에 대한 세부 정보를 표시합니다.

```
aws outposts get-outpost \
    --outpost-id op-0ab23c4567EXAMPLE
```

출력:

```
{
    "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": {}
    }
}
```

자세한 내용은 Outposts 사용 설명서의 Outposts 작업을 AWS 참조하세요.

- 자세한 API 내용은 명령 참조GetOutpost의 섹션을 참조하세요. AWS CLI

### `list-outposts`

다음 코드 예시에서는 `list-outposts`을 사용하는 방법을 보여 줍니다.

AWS CLI

Outposts를 나열하려면

다음 `list-outposts` 예제에서는 AWS 계정의 Outpost를 나열합니다.

```
aws outposts list-outposts
```

출력:

```
{
    "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": {}
        }
    ]
}
```

자세한 내용은 Outposts 사용 설명서의 Outposts 작업을 AWS 참조하세요.

- 자세한 API 내용은 명령 참조ListOutposts의 섹션을 참조하세요. AWS CLI

**`list-sites`**

다음 코드 예시에서는 `list-sites`을 사용하는 방법을 보여 줍니다.

AWS CLI

사이트를 나열하려면

다음 `list-sites` 예제에서는 AWS 계정에서 사용 가능한 Outpost 사이트를 나열합니다.

```
aws outposts list-sites
```

출력:

```
{
    "Sites": [
        {
            "SiteId": "os-0ab12c3456EXAMPLE",
            "AccountId": "123456789012",
            "Name": "EXAMPLE",
            "Description": "example",
            "Tags": {}
        }
    ]
}
```

자세한 내용은 Outposts 사용 설명서의 Outposts 작업을 AWS 참조하세요.

- 자세한 API 내용은 명령 참조ListSites의 섹션을 참조하세요. AWS CLI

## AWS Payment Cryptography 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS Payment Cryptography.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `create-alias`

다음 코드 예시에서는 `create-alias`을 사용하는 방법을 보여 줍니다.

AWS CLI

키의 별칭을 생성하려면

다음 `create-alias` 예제에서는 키의 별칭을 생성합니다.

```
aws payment-cryptography create-alias \
    --alias-name alias/sampleAlias1 \
    --key-arn arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h
```

출력:

```
{
    "Alias": {
        "AliasName": "alias/sampleAlias1",
        "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/
kwapwa6qaifllw2h"
    }
}
```

자세한 내용은 AWS Payment Cryptography 사용 설명서의 별칭 정보를 참조하세요.

- 자세한 API 내용은 명령 참조CreateAlias의 섹션을 참조하세요. AWS CLI

### `create-key`

다음 코드 예시에서는 `create-key`을 사용하는 방법을 보여 줍니다.

AWS CLI

키를 생성하려면

다음 create-key 예제에서는 CVV/CVV2 값을 생성하고 확인하는 데 사용할 수 있는 2KEY개의 TDES 키를 생성합니다.

```
aws payment-cryptography create-key \
    --exportable \
    --key-
attributes KeyAlgorithm=TDES_2KEY, KeyUsage=TR31_C0_CARD_VERIFICATION_KEY,KeyClass=SYMMETRIC
```

출력:

```
{
    "Key": {
        "CreateTimestamp": "1686800690",
        "Enabled": true,
        "Exportable": true,
        "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/
kwapwa6qaifllw2h",
        "KeyAttributes": {
            "KeyAlgorithm": "TDES_2KEY",
            "KeyClass": "SYMMETRIC_KEY",
            "KeyModesOfUse": {
                "Decrypt": false,
                "DeriveKey": false,
                "Encrypt": false,
                "Generate": true,
                "NoRestrictions": false,
                "Sign": false,
                "Unwrap": false,
                "Verify": true,
                "Wrap": false
            },
            "KeyUsage": "TR31_C0_CARD_VERIFICATION_KEY"
        },
        "KeyCheckValue": "F2E50F",
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "KeyState": "CREATE_COMPLETE",
        "UsageStartTimestamp": "1686800690"
    }
}
```

자세한 내용은 AWS Payment Cryptography 사용 설명서의 키 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateKey의 섹션을 참조하세요. AWS CLI

### `delete-alias`

다음 코드 예시에서는 `delete-alias`을 사용하는 방법을 보여 줍니다.

AWS CLI

별칭을 삭제하려면

다음 `delete-alias` 예제에서는 별칭을 삭제합니다. 키에는 영향을 주지 않습니다.

```
aws payment-cryptography delete-alias \
    --alias-name alias/sampleAlias1
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Payment Cryptography 사용 설명서의 별칭 정보를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteAlias의 섹션을 참조하세요. AWS CLI

### `delete-key`

다음 코드 예시에서는 `delete-key`을 사용하는 방법을 보여 줍니다.

AWS CLI

키를 삭제하려면

다음 `delete-key` 예제에서는 기본 대기 기간인 7일 후에 삭제할 키를 예약합니다.

```
aws payment-cryptography delete-key \
    --key-identifier arn:aws:payment-cryptography:us-west-2:123456789012:key/
kwapwa6qaifllw2h
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS Payment Cryptography 사용 설명서의 키 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteKey의 섹션을 참조하세요. AWS CLI

### `export-key`

다음 코드 예시에서는 `export-key`을 사용하는 방법을 보여 줍니다.

AWS CLI

키를 내보내려면

다음 `export-key` 예제에서는 키를 내보냅니다.

```
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 }}'
```

wrapping-key-certificate.pem의 콘텐츠:

```
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUV2VENDQXFXZ0F3SUJBZ0lSQU1ZZS8xMXFUK2svVzlRUDJQOElV
```

출력:

```
{
    "WrappedKey": {
        "KeyMaterial":
 "308205A106092A864886F70D010702A08205923082058E020101310D300B0609608648016503040201308203F
        "WrappedKeyMaterialFormat": "TR34_KEY_BLOCK"
    }
}
```

자세한 내용은 AWS Payment Cryptography 사용 설명서의 키 내보내기를 참조하세요.

- 자세한 API 내용은 명령 참조ExportKey의 섹션을 참조하세요. AWS CLI

### get-alias

다음 코드 예시에서는 get-alias을 사용하는 방법을 보여 줍니다.

AWS CLI

별칭을 가져오려면

다음 get-alias 예제에서는 별칭과 연결된 키ARN의 를 반환합니다.

```
aws payment-cryptography get-alias \
    --alias-name alias/sampleAlias1
```

출력:

```
{
    "Alias": {
        "AliasName": "alias/sampleAlias1",
        "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/
kwapwa6qaifllw2h"
    }
}
```

자세한 내용은 AWS Payment Cryptography 사용 설명서의 별칭 정보를 참조하세요.

- 자세한 API 내용은 명령 참조GetAlias의 섹션을 참조하세요. AWS CLI

### get-key

다음 코드 예시에서는 get-key을 사용하는 방법을 보여 줍니다.

AWS CLI

키의 메타데이터를 가져오려면

다음 get-key 예제에서는 별칭과 연결된 키의 메타데이터를 반환합니다. 이 작업은 암호화 자료를 반환하지 않습니다.

```
aws payment-cryptography get-key \
    --key-identifier alias/sampleAlias1
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS Payment Cryptography 사용 설명서의 키 가져오기를 참조하세요.

- 자세한 API 내용은 명령 참조GetKey의 섹션을 참조하세요. AWS CLI

### get-parameters-for-export

다음 코드 예시에서는 get-parameters-for-export을 사용하는 방법을 보여 줍니다.

AWS CLI

내보내기 프로세스를 초기화하려면

다음 get-parameters-for-export 예제에서는 키 페어를 생성하고 키에 서명한 다음 인증서와 인증서 루트를 반환합니다.

```
aws payment-cryptography get-parameters-for-export \
    --signing-key-algorithm RSA_2048 \
    --key-material-type TR34_KEY_BLOCK
```

출력:

```
{
    "ExportToken": "export-token-ep5cwyzune7oya53",
    "ParametersValidUntilTimestamp": "1687415640",
    "SigningKeyAlgorithm": "RSA_2048",
```

```
    "SigningKeyCertificate":

    "MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
    VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
    b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
    BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
    MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
    VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
    b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
    YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
    21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
    rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
    Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
    nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
    FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
    NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=",
    "SigningKeyCertificateChain":
    "NIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
    VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
    b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
    BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
    MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
    VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
    b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
    YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
    21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
    rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
    Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
    nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
    FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
    NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE="
}
```

자세한 내용은 AWS Payment Cryptography 사용 설명서의 키 내보내기를 참조하세요.

- 자세한 API 내용은 명령 참조GetParametersForExport의 섹션을 참조하세요. AWS CLI

### `get-parameters-for-import`

다음 코드 예시에서는 `get-parameters-for-import`을 사용하는 방법을 보여 줍니다.

AWS CLI

가져오기 프로세스를 초기화하려면

다음 get-parameters-for-import 예제에서는 키 페어를 생성하고 키에 서명한 다음 인증서와 인증서 루트를 반환합니다.

```
aws payment-cryptography get-parameters-for-import \
    --key-material-type TR34_KEY_BLOCK \
    --wrapping-key-algorithm RSA_2048
```

출력:

```
{
    "ImportToken": "import-token-qgmafpaa7nt2kfbb",
    "ParametersValidUntilTimestamp": "1687415640",
    "WrappingKeyAlgorithm": "RSA_2048",
    "WrappingKeyCertificate":
    "MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
    VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
    b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
    BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
    MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
    VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
    b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
    YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
    21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
    rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
    Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
    nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
    FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
    NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=",
    "WrappingKeyCertificateChain":
    "NIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
    VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
    b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
    BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
    MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
    VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
    b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
    YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
    21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
    rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
    Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
    nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
    FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
    NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE="
```

```
}
```

자세한 내용은 AWS Payment Cryptography 사용 설명서의 키 가져오기를 참조하세요.

- 자세한 API 내용은 명령 참조GetParametersForImport의 섹션을 참조하세요. AWS CLI

### get-public-key-certificate

다음 코드 예시에서는 get-public-key-certificate을 사용하는 방법을 보여 줍니다.

AWS CLI

퍼블릭 키를 반환하려면

다음 get-public-key-certificate 예제에서는 키 페어의 퍼블릭 키 부분을 반환합니다.

```
aws payment-cryptography get-public-key-certificate \
    --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h
```

출력:

```
{
    "KeyCertificate":
    "MIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
    VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
    b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
    BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
    MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
    VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
    b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
    YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
    21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
    rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
    Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
    nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
    FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
    NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=",
    "KeyCertificateChain":
    "NIICiTCCAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
    VVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
    b24xFDASBgNVBAsTC0lBTSBDb25zb2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
```

```
    BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhcNMTEwNDI1MjA0NTIxWhcN
    MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
    VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
    b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
    YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
    21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
    rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
    Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
    nUhVVxYUntneD9+h8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
    FFBjvSfpJIlJ00zbhNYS5f6GuoEDmFJl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
    NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE="
}
```

자세한 내용은 AWS Payment Cryptography 사용 설명서의 키 페어와 연결된 퍼블릭 키/인증서 가져오기를 참조하세요.

- 자세한 API 내용은 명령 참조GetPublicKeyCertificate의 섹션을 참조하세요. AWS CLI

### `import-key`

다음 코드 예시에서는 `import-key`을 사용하는 방법을 보여 줍니다.

AWS CLI

TR-34 키를 가져오려면

다음 `import-key` 예제에서는 TR-34 키를 가져옵니다.

```
aws payment-cryptography import-key \
        --key-material='{ "Tr34KeyBlock": {" \
            CertificateAuthorityPublicKeyIdentifier": "arn:aws:payment-
cryptography:us-west-2:123456789012:key/rmm5wn2q564njnjm", \
            "ImportToken": "import-token-5ott6ho5nts7bbcg", \
            "KeyBlockFormat": "X9_TR34_2012", \
            "SigningKeyCertificate": file://signing-key-certificate.pem, \
            "WrappedKeyBlock": file://wrapped-key-block.pem }}'
```

`signing-key-certificate.pem`의 콘텐츠:

```
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUV2RENDQXFTZ0F3SUJBZ0lRYWVCK25IbE1WZU1PR1ZiNjU1Q2Jz
```

`wrapped-key-block.pem`의 콘텐츠:

```
3082059806092A864886F70D010702A0820589308205850201013 10D300B06096086480165030402013082031606
```

출력:

```
{
    "Key": {
        "CreateTimestamp": "2023-06-09T16:56:27.621000-07:00",
        "Enabled": true,
        "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/
bzmvgyxdg3sktwxd",
        "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"
    }
}
```

자세한 내용은 AWS Payment Cryptography 사용 설명서의 키 가져오기를 참조하세요.

- 자세한 API 내용은 명령 참조ImportKey의 섹션을 참조하세요. AWS CLI

### `list-aliases`

다음 코드 예시에서는 `list-aliases`을 사용하는 방법을 보여 줍니다.

AWS CLI

별칭 목록을 가져오려면

다음 `list-aliases` 예제는 이 리전의 계정에 있는 모든 별칭을 보여줍니다.

```
aws payment-cryptography list-aliases
```

출력:

```
{
    "Aliases": [
        {
            "AliasName": "alias/sampleAlias1",
            "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h"
        },
        {
            "AliasName": "alias/sampleAlias2",
            "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h"
        }
    ]
}
```

자세한 내용은 AWS Payment Cryptography 사용 설명서의 별칭 정보를 참조하세요.

- 자세한 API 내용은 명령 참조ListAliases의 섹션을 참조하세요. AWS CLI

## list-keys

다음 코드 예시에서는 `list-keys`을 사용하는 방법을 보여 줍니다.

AWS CLI

키 목록을 가져오려면

다음 `list-keys` 예제에서는 이 리전의 계정에 있는 모든 키를 보여줍니다.

```
aws payment-cryptography list-keys
```

출력:

```
{
    "Keys": [
    {
        "CreateTimestamp": "1666506840",
        "Enabled": false,
        "Exportable": true,
        "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h",
        "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"
    }
    ]
}
```

자세한 내용은 AWS Payment Cryptography 사용 설명서의 키 나열을 참조하세요.

- 자세한 API 내용은 명령 참조ListKeys의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

키의 태그 목록을 가져오려면

다음 `list-tags-for-resource` 예제에서는 키에 대한 태그를 가져옵니다.

```
aws payment-cryptography list-tags-for-resource \
    --resource-arn arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h
```

출력:

```
{
    "Tags": [
        {
            "Key": "BIN",
            "Value": "20151120"
        },
        {
            "Key": "Project",
            "Value": "Production"
        }
    ]
}
```

자세한 내용은 AWS Payment Cryptography 사용 설명서의 API 작업을 사용하여 키 태그 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

## `restore-key`

다음 코드 예시에서는 `restore-key`을 사용하는 방법을 보여 줍니다.

AWS CLI

삭제가 예약된 키를 복원하려면

다음 `restore-key` 예제에서는 키 삭제를 취소합니다.

```
aws payment-cryptography restore-key \
```

```
    --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h
```

출력:

```
{
    "Key": {
        "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h",
        "KeyAttributes": {
            "KeyUsage": "TR31_V2_VISA_PIN_VERIFICATION_KEY",
            "KeyClass": "SYMMETRIC_KEY",
            "KeyAlgorithm": "TDES_3KEY",
            "KeyModesOfUse": {
                "Encrypt": false,
                "Decrypt": false,
                "Wrap": false,
                "Unwrap": false,
                "Generate": true,
                "Sign": false,
                "Verify": true,
                "DeriveKey": false,
                "NoRestrictions": false
            }
        },
        "KeyCheckValue": "",
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
        "Enabled": false,
        "Exportable": true,
        "KeyState": "CREATE_COMPLETE",
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "CreateTimestamp": "1686800690",
        "UsageStopTimestamp": "1687405998"
    }
}
```

자세한 내용은 AWS Payment Cryptography 사용 설명서의 키 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조RestoreKey의 섹션을 참조하세요. AWS CLI

### `start-key-usage`

다음 코드 예시에서는 `start-key-usage`을 사용하는 방법을 보여 줍니다.

AWS CLI

키를 활성화하려면

다음 `start-key-usage` 예제에서는 키를 사용할 수 있습니다.

```
aws payment-cryptography start-key-usage \
    --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS Payment Cryptography 사용 설명서의 키 활성화 및 비활성화를 참조하세요.

- 자세한 API 내용은 명령 참조StartKeyUsage의 섹션을 참조하세요. AWS CLI

### stop-key-usage

다음 코드 예시에서는 stop-key-usage을 사용하는 방법을 보여 줍니다.

AWS CLI

키를 비활성화하려면

다음 stop-key-usage 예제에서는 키를 비활성화합니다.

```
aws payment-cryptography stop-key-usage \
    --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS Payment Cryptography 사용 설명서의 키 활성화 및 비활성화를 참조하세요.

- 자세한 API 내용은 명령 참조StopKeyUsage의 섹션을 참조하세요. AWS CLI

### `tag-resource`

다음 코드 예시에서는 `tag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

키에 태그를 지정하려면

다음 `tag-resource` 예제에서는 키에 태그를 지정합니다.

```
aws payment-cryptography tag-resource \
    --resource-arn arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h \
    --tags Key=sampleTag,Value=sampleValue
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Payment Cryptography 사용 설명서의 키 태그 관리를 참조하세요.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### `untag-resource`

다음 코드 예시에서는 `untag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

키에서 태그를 제거하려면

다음 `untag-resource` 예제에서는 키에서 태그를 제거합니다.

```
aws payment-cryptography untag-resource \
    --resource-arn arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h \
    --tag-keys sampleTag
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Payment Cryptography 사용 설명서의 키 태그 관리를 참조하세요.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-alias

다음 코드 예시에서는 `update-alias`을 사용하는 방법을 보여 줍니다.

AWS CLI

별칭을 업데이트하려면

다음 `update-alias` 예제에서는 별칭을 다른 키와 연결합니다.

```
aws payment-cryptography update-alias \
    --alias-name alias/sampleAlias1 \
    --key-arn arn:aws:payment-cryptography:us-east-2:123456789012:key/
tqv5yij6wtxx64pi
```

출력:

```
{
    "Alias": {
        "AliasName": "alias/sampleAlias1",
        "KeyArn": "arn:aws:payment-cryptography:us-west-2:123456789012:key/
tqv5yij6wtxx64pi "
    }
}
```

자세한 내용은 AWS Payment Cryptography 사용 설명서의 별칭 정보를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateAlias의 섹션을 참조하세요. AWS CLI

# AWS Payment Cryptography 를 사용한 데이터 플레인 예제 AWS CLI

다음 코드 예제에서는 AWS Payment Cryptography Data Plane과 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `decrypt-data`

다음 코드 예시에서는 `decrypt-data`을 사용하는 방법을 보여 줍니다.

AWS CLI

암호 텍스트를 복호화하려면

다음 `decrypt-data` 예제에서는 대칭 키를 사용하여 암호 텍스트 데이터를 복호화합니다. 이 작업을 수행하려면 키가 로 `KeyModesOfUse` 설정되어 `Decrypt` 있고 로 `KeyUsage` 설정되어 있어야 합니다`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}'
```

출력:

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h",
```

```
    "KeyCheckValue": "71D7AE",
    "PlainText": "31323334313233343132333431323334"
}
```

자세한 내용은 AWS Payment Cryptography 사용 설명서의 데이터 복호화를 참조하세요.

- 자세한 API 내용은 명령 참조DecryptData의 섹션을 참조하세요. AWS CLI

### encrypt-data

다음 코드 예시에서는 encrypt-data을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터를 암호화하려면

다음 encrypt-data 예제에서는 대칭 키를 사용하여 일반 텍스트 데이터를 암호화합니다. 이 작업을 수행하려면 키가 로 KeyModesOfUse 설정되어 Encrypt 있고 로 KeyUsage 설정되어 있어야 합니다TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY.

```
aws payment-cryptography-data encrypt-data \
    --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h \
    --plain-text 31323334313233343132333431323334 \
    --encryption-attributes 'Symmetric={Mode=CBC}'
```

출력:

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h",
    "KeyCheckValue": "71D7AE",
    "CipherText": "33612AB9D6929C3A828EB6030082B2BD"
}
```

자세한 내용은 AWS Payment Cryptography 사용 설명서의 데이터 암호화를 참조하세요.

- 자세한 API 내용은 명령 참조EncryptData의 섹션을 참조하세요. AWS CLI

### generate-card-validation-data

다음 코드 예시에서는 generate-card-validation-data을 사용하는 방법을 보여 줍니다.

AWS CLI

를 생성하려면 CVV

다음 generate-card-validation-data 예제에서는 CVV/를 생성합니다CVV2.

```
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}
```

출력:

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h",
    "KeyCheckValue": "CADDA1",
    "ValidationData": "801"
}
```

자세한 내용은 AWS Payment Cryptography 사용 설명서의 카드 데이터 생성을 참조하세요.

- 자세한 API 내용은 명령 참조GenerateCardValidationData의 섹션을 참조하세요. AWS CLI

### generate-mac

다음 코드 예시에서는 generate-mac을 사용하는 방법을 보여 줍니다.

AWS CLI

를 생성하려면 MAC

다음 generate-card-validation-data 예제에서는 알고리즘 HMAC_SHA256 및 HMAC 암호화 키를 사용하여 카드 데이터 인증을 위한 해시 기반 메시지 인증 코드(HMAC)를 생성합니다. 키는 TR31_M7_HMAC_KEY 로, KeyModesOfUse로 KeyUsage 설정되어 있어야 합니다Generate.

```
aws payment-cryptography-data generate-mac \
    --key-identifier arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h \
    --message-
data "3b3130383834393030313037333934313533d32343038323236303030373030303f33" \
```

```
    --generation-attributes Algorithm=HMAC_SHA256
```

출력:

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:123456789012:key/
kwapwa6qaifllw2h,
    "KeyCheckValue": "2976E7",
    "Mac": "ED87F26E961C6D0DDB78DA5038AA2BDDEA0DCE03E5B5E96BDDD494F4A7AA470C"
}
```

자세한 내용은 AWS Payment Cryptography 사용 설명서의 생성을 MAC 참조하세요.

- 자세한 API 내용은 명령 참조GenerateMac의 섹션을 참조하세요. AWS CLI

### generate-pin-data

다음 코드 예시에서는 generate-pin-data을 사용하는 방법을 보여 줍니다.

AWS CLI

를 생성하려면 PIN

다음 generate-card-validation-data 예제에서는 Visa PIN 체계를 PIN 사용하여 새 무작위를 생성합니다.

```
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}
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS Payment Cryptography 사용 설명서의 PIN 데이터 생성을 참조하세요.

- 자세한 API 내용은 명령 참조GeneratePinData의 섹션을 참조하세요. AWS CLI

### re-encrypt-data

다음 코드 예시에서는 re-encrypt-data을 사용하는 방법을 보여 줍니다.

AWS CLI

다른 키로 데이터를 다시 암호화하려면

다음 re-encrypt-data 예제에서는 AES 대칭 키를 사용하여 암호화된 암호 텍스트를 복호화하고 트랜잭션당 파생된 고유 키(DUKPT) 키를 사용하여 다시 암호화합니다.

```
aws payment-cryptography-data re-encrypt-data \
    --incoming-key-identifier arn:aws:payment-cryptography:us-
west-2:111122223333:key/hyvv7ymboitd4vfy \
    --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"}}'
```

출력:

```
{
    "CipherText":
 "F94959DA30EEFF0C035483C6067667CF6796E3C1AD28C2B61F9CFEB772A8DD41C0D6822931E0D3B1",
    "KeyArn": "arn:aws:payment-cryptography:us-west-2:111122223333:key/
jl6ythkcvzesbxen",
```

```
    "KeyCheckValue": "2E8CD9"
}
```

자세한 내용은 AWS Payment Cryptography 사용 설명서의 데이터 암호화 및 복호화를 참조하세요.

- 자세한 API 내용은 명령 참조ReEncryptData의 섹션을 참조하세요. AWS CLI

### `translate-pin-data`

다음 코드 예시에서는 `translate-pin-data`을 사용하는 방법을 보여 줍니다.

AWS CLI

PIN 데이터를 번역하려면

다음 `translate-pin-data` 예제에서는 를 ISO 0 PIN 블록을 사용하는 PEK TDES 암호화PIN에서 DUKPT 알고리즘을 사용하는 AES ISO 4 PIN 블록으로 변환합니다.

```
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"
```

출력:

```
{
    "PinBlock": "1F4209C670E49F83E75CC72E81B787D9",
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
ivi5ksfsuplneuyt
    "KeyCheckValue": "7CC9E2"
}
```

자세한 내용은 AWS Payment Cryptography 사용 설명서의 PIN 데이터 번역을 참조하세요.

- 자세한 API 내용은 명령 참조TranslatePinData의 섹션을 참조하세요. AWS CLI

### verify-auth-request-cryptogram

다음 코드 예시에서는 verify-auth-request-cryptogram을 사용하는 방법을 보여 줍니다.

AWS CLI

인증 요청을 확인하려면

다음 verify-auth-request-cryptogram 예제에서는 권한 부여 요청 암호()를 확인합니다 ARQC.

```
aws payment-cryptography-data verify-auth-request-cryptogram \
    --auth-request-cryptogram F6E1BD1E6037FB3E \
    --auth-response-attributes '{"ArpcMethod1": {"AuthResponseCode": "1111"}}' \
    --key-identifier arn:aws:payment-cryptography:us-west-2:111122223333:key/
pboipdfzd4mdklya \
    --major-key-derivation-mode "EMV_OPTION_A" \
    --session-key-derivation-attributes '{"EmvCommon":
 {"ApplicationTransactionCounter": "1234","PanSequenceNumber":
 "01","PrimaryAccountNumber": "471234567890123"}}' \
    --transaction-data "123456789ABCDEF"
```

출력:

```
{
    "AuthResponseValue": "D899B8C6FBF971AA",
    "KeyArn": "arn:aws:payment-cryptography:us-west-2:111122223333:key/
pboipdfzd4mdklya",
    "KeyCheckValue": "985792"
}
```

자세한 내용은 AWS Payment Cryptography 사용 설명서의 인증 요청 확인(ARQC) 암호를 참조하세요.

- 자세한 API 내용은 명령 참조VerifyAuthRequestCryptogram의 섹션을 참조하세요. AWS CLI

### verify-card-validation-data

다음 코드 예시에서는 verify-card-validation-data을 사용하는 방법을 보여 줍니다.

AWS CLI

를 검증하려면 CVV

다음 `verify-card-validation-data` 예제에서는 에 대해 CVV/CVV2를 검증합니다PAN.

```
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
```

출력:

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
tqv5yij6wtxx64pi",
    "KeyCheckValue": "CADDA1"
}
```

자세한 내용은 AWS Payment Cryptography 사용 설명서의 카드 데이터 확인을 참조하세요.

- 자세한 API 내용은 명령 참조VerifyCardValidationData의 섹션을 참조하세요. AWS CLI

### verify-mac

다음 코드 예시에서는 `verify-mac`을 사용하는 방법을 보여 줍니다.

AWS CLI

를 확인하려면 MAC

다음 `verify-mac` 예제에서는 알고리즘 HMAC_SHA256 및 HMAC 암호화 키를 사용하여 카드 데이터 인증을 위한 해시 기반 메시지 인증 코드(HMAC)를 확인합니다.

```
aws payment-cryptography-data verify-mac \
    --key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
qnobl5lghrzunce6 \
    --message-
data "3b34303838343930303130373333934313533d323430383232363030303730303033f33" \
    --verification-attributes='Algorithm=HMAC_SHA256' \
```

```
    --mac ED87F26E961C6D0DDB78DA5038AA2BDDEA0DCE03E5B5E96BDDD494F4A7AA470C
```

출력:

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
qnobl5lghrzunce6,
    "KeyCheckValue": "2976E7",
}
```

자세한 내용은 AWS Payment Cryptography 사용 설명서의 확인을 MAC 참조하세요.

- 자세한 API 내용은 명령 참조VerifyMac의 섹션을 참조하세요. AWS CLI

### verify-pin-data

다음 코드 예시에서는 verify-pin-data을 사용하는 방법을 보여 줍니다.

AWS CLI

를 확인하려면 PIN

다음 verify-pin-data 예제에서는 에 PIN 대한 를 검증합니다PAN.

```
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
```

출력:

```
{
    "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",
}
```

자세한 내용은 AWS Payment Cryptography 사용 설명서의 PIN 데이터 확인을 참조하세요.

- 자세한 API 내용은 명령 참조VerifyPinData의 섹션을 참조하세요. AWS CLI

## 를 사용한 Amazon Pinpoint 예제 AWS CLI

다음 코드 예제에서는 Amazon Pinpoint 와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `create-app`

다음 코드 예시에서는 `create-app`을 사용하는 방법을 보여 줍니다.

AWS CLI

예시 1: 애플리케이션 생성

다음 `create-app` 예시에서는 새 애플리케이션(프로젝트)을 생성합니다.

```
aws pinpoint create-app \
    --create-application-request Name=ExampleCorp
```

출력:

```
{
    "ApplicationResponse": {
        "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
        "Id": "810c7aab86d42fb2b56c8c966example",
        "Name": "ExampleCorp",
        "tags": {}
    }
}
```

예시 2: 태그가 지정된 애플리케이션을 생성하는 방법

다음 `create-app` 예시에서는 새 애플리케이션(프로젝트)을 만들고 태그(키 및 값)를 애플리케이션에 연결합니다.

```
aws pinpoint create-app \
    --create-application-request Name=ExampleCorp,tags={"Stack"="Test"}
```

출력:

```
{
    "ApplicationResponse": {
        "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
        "Id": "810c7aab86d42fb2b56c8c966example",
        "Name": "ExampleCorp",
        "tags": {
            "Stack": "Test"
        }
    }
}
```

- 자세한 API 내용은 명령 참조CreateApp의 섹션을 참조하세요. AWS CLI

### `create-sms-template`

다음 코드 예시에서는 `create-sms-template`을 사용하는 방법을 보여 줍니다.

AWS CLI

SMS 채널을 통해 전송되는 메시지에 대한 메시지 템플릿을 생성합니다.

다음 create-sms-template 예제에서는 SMS 메시지 템플릿을 생성합니다.

```
aws pinpoint create-sms-template \
    --template-name TestTemplate \
    --sms-template-request file://myfile.json \
    --region us-east-1
```

myfile.json의 콘텐츠:

```
{
    "Body": "hello\n how are you?\n food is good",
    "TemplateDescription": "Test SMS Template"
}
```

출력:

```
{
    "CreateTemplateMessageBody": {
        "Arn": "arn:aws:mobiletargeting:us-east-1:AIDACKCEVSQ6C2EXAMPLE:templates/
TestTemplate/SMS",
        "Message": "Created",
        "RequestID": "8c36b17f-a0b0-400f-ac21-29e9b62a975d"
    }
}
```

자세한 내용은 Amazon Pinpoint 사용 설명서의 Amazon Pinpoint 메시지 템플릿을 참조하세요. Amazon Pinpoint

- 자세한 API 내용은 명령 참조CreateSmsTemplate의 섹션을 참조하세요. AWS CLI

### delete-app

다음 코드 예시에서는 delete-app을 사용하는 방법을 보여 줍니다.

AWS CLI

애플리케이션 삭제

다음 delete-app 예시에서는 애플리케이션(프로젝트)을 삭제합니다.

```
aws pinpoint delete-app \
```

```
    --application-id 810c7aab86d42fb2b56c8c966example
```

출력:

```
{
    "ApplicationResponse": {
        "Arn": "arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
        "Id": "810c7aab86d42fb2b56c8c966example",
        "Name": "ExampleCorp",
        "tags": {}
    }
}
```

- 자세한 API 내용은 명령 참조DeleteApp의 섹션을 참조하세요. AWS CLI

### get-apns-channel

다음 코드 예시에서는 get-apns-channel을 사용하는 방법을 보여 줍니다.

AWS CLI

애플리케이션 APNs 채널의 상태 및 설정에 대한 정보를 검색하려면

다음 get-apns-channel 예제에서는 애플리케이션의 APNs 채널 상태 및 설정에 대한 정보를 검색합니다.

```
aws pinpoint get-apns-channel \
    --application-id 9ab1068eb0a6461c86cce7f27ce0efd7 \
    --region us-east-1
```

출력:

```
{
    "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
    }
}
```

- 자세한 API 내용은 명령 참조GetApnsChannel의 섹션을 참조하세요. AWS CLI

### get-app

다음 코드 예시에서는 get-app을 사용하는 방법을 보여 줍니다.

AWS CLI

애플리케이션에 대한 정보를 검색하려면(프로젝트)

다음 get-app 예제에서는 애플리케이션(프로젝트)에 대한 정보를 검색합니다.

```
aws pinpoint get-app \
    --application-id 810c7aab86d42fb2b56c8c966example \
    --region us-east-1
```

출력:

```
{
    "ApplicationResponse": {
        "Arn": "arn:aws:mobiletargeting:us-
east-1:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example",
        "Id": "810c7aab86d42fb2b56c8c966example",
        "Name": "ExampleCorp",
        "tags": {
                "Year": "2019",
                "Stack": "Production"
            }
    }
}
```

- 자세한 API 내용은 명령 참조GetApp의 섹션을 참조하세요. AWS CLI

### get-apps

다음 코드 예시에서는 get-apps을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 애플리케이션에 대한 정보를 검색하려면

다음 get-apps 예제에서는 모든 애플리케이션(프로젝트)에 대한 정보를 검색합니다.

```
aws pinpoint get-apps
```

출력:

```
{
    "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":
 "eyJDcmVhdGlvbkRhdGUiOiIyMDE5LTA3LTE2VDE0OjM4OjUzLjkwM1oiLCJBY2NvdW50SWQiOiI1MTIzOTcxODM4Nz
    }
}
```

NextToken 응답 값의 존재는 사용 가능한 출력이 더 많음을 나타냅니다. 명령을 다시 호출하고 해당 값을 NextToken 입력 파라미터로 입력합니다.

- 자세한 API 내용은 명령 참조GetApps의 섹션을 참조하세요. AWS CLI

### get-campaign

다음 코드 예시에서는 get-campaign을 사용하는 방법을 보여 줍니다.

AWS CLI

캠페인의 상태, 구성 및 기타 설정에 대한 정보를 검색하려면

다음 get-campaign 예제에서는 캠페인의 상태, 구성 및 기타 설정에 대한 정보를 검색합니다.

```
aws pinpoint get-campaign \
    --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
    --campaign-id a1e63c6cc0eb43ed826ffcc3cc90b30d \
    --region us-east-1
```

출력:

```
{
    "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
    }
}
```

- 자세한 API 내용은 명령 참조GetCampaign의 섹션을 참조하세요. AWS CLI

### `get-campaigns`

다음 코드 예시에서는 `get-campaigns`을 사용하는 방법을 보여 줍니다.

AWS CLI

애플리케이션과 연결된 모든 캠페인의 상태, 구성 및 기타 설정에 대한 정보를 검색하려면

다음 `get-campaigns` 예제에서는 애플리케이션과 연결된 모든 캠페인의 상태, 구성 및 기타 설정에 대한 정보를 검색합니다.

```
aws pinpoint get-campaigns \
    --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
    --region us-east-1
```

출력:

```
{
    "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=\"en
\">\n    <head>\n    <meta http-equiv=\"Content-Type\" content=\"text/html;
 charset=utf-8\" />\n</head>\n<body>Demo</body>\n</html>",
                        "Title": "PinpointDemo"
                    }
                },
                "Name": "MyCampaign2",
                "Schedule": {
                    "IsLocalTime": false,
                    "StartTime": "IMMEDIATE",
                    "Timezone": "utc"
                },
                "SegmentId": "b66c9e42f71444b2aa2e0ffc1df28f60",
                "SegmentVersion": 1,
                "State": {
                    "CampaignStatus": "COMPLETED"
                },
                "tags": {},
                "TemplateConfiguration": {},
```

```
                "Version": 1
            }
        ]
    }
}
```

- 자세한 API 내용은 명령 참조GetCampaigns의 섹션을 참조하세요. AWS CLI

### `get-channels`

다음 코드 예시에서는 `get-channels`을 사용하는 방법을 보여 줍니다.

AWS CLI

애플리케이션에 대한 각 채널의 기록 및 상태에 대한 정보를 검색하려면

다음 `get-channels` 예제에서는 애플리케이션에 대한 각 채널의 기록 및 상태에 대한 정보를 검색합니다.

```
aws pinpoint get-channels \
    --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
    --region us-east-1
```

출력:

```
{
    "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
            }
        }
    }
}
```

- 자세한 API 내용은 명령 참조GetChannels의 섹션을 참조하세요. AWS CLI

### `get-email-channel`

다음 코드 예시에서는 `get-email-channel`을 사용하는 방법을 보여 줍니다.

AWS CLI

애플리케이션의 이메일 채널 상태 및 설정에 대한 정보를 검색하려면

다음 `get-email-channel` 예제에서는 애플리케이션에 대한 이메일 채널의 상태 및 설정을 검색합니다.

```
aws pinpoint get-email-channel \
    --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
    --region us-east-1
```

출력:

```
{
    "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
    }
}
```

- 자세한 API 내용은 명령 참조GetEmailChannel의 섹션을 참조하세요. AWS CLI

### `get-endpoint`

다음 코드 예시에서는 `get-endpoint`을 사용하는 방법을 보여 줍니다.

AWS CLI

애플리케이션에 대한 특정 엔드포인트의 설정 및 속성에 대한 정보 검색

다음 `get-endpoint` 예시에서는 애플리케이션에 대한 특정 엔드포인트의 설정 및 속성에 대한 정보를 검색합니다.

```
aws pinpoint get-endpoint \
    --application-id 611e3e3cdd47474c9c1399a505665b91 \
    --endpoint-id testendpoint \
    --region us-east-1
```

출력:

```
{
    "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"
        }
    }
}
```

- 자세한 API 내용은 명령 참조GetEndpoint의 섹션을 참조하세요. AWS CLI

### `get-gcm-channel`

다음 코드 예시에서는 `get-gcm-channel`을 사용하는 방법을 보여 줍니다.

AWS CLI

애플리케이션 GCM 채널의 상태 및 설정에 대한 정보를 검색하려면

다음 `get-gcm-channel` 예제에서는 애플리케이션의 GCM 채널 상태 및 설정에 대한 정보를 검색합니다.

```
aws pinpoint get-gcm-channel \
    --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
    --region us-east-1
```

출력:

```
{
    "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
    }
}
```

- 자세한 API 내용은 명령 참조GetGcmChannel의 섹션을 참조하세요. AWS CLI

### get-sms-channel

다음 코드 예시에서는 get-sms-channel을 사용하는 방법을 보여 줍니다.

AWS CLI

애플리케이션 SMS 채널의 상태 및 설정에 대한 정보를 검색하려면

다음 get-sms-channel 예시에서는 애플리케이션의 SMS 채널 상태 및 설정을 검색합니다.

```
aws pinpoint get-sms-channel \
    --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
    --region us-east-1
```

출력:

```
{
    "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
    }
}
```

- 자세한 API 내용은 명령 참조GetSmsChannel의 섹션을 참조하세요. AWS CLI

### get-sms-template

다음 코드 예시에서는 get-sms-template을 사용하는 방법을 보여 줍니다.

AWS CLI

SMS 채널을 통해 전송되는 메시지에 대한 메시지 템플릿의 콘텐츠 및 설정을 검색합니다.

다음 get-sms-template 예제에서는 SMS 메시지 템플릿의 콘텐츠와 설정을 검색합니다.

```
aws pinpoint get-sms-template \
    --template-name TestTemplate \
    --region us-east-1
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Pinpoint 사용 설명서의 Amazon Pinpoint 메시지 템플릿을 참조하세요. Amazon Pinpoint

- 자세한 API 내용은 명령 참조GetSmsTemplate의 섹션을 참조하세요. AWS CLI

### `get-voice-channel`

다음 코드 예시에서는 `get-voice-channel`을 사용하는 방법을 보여 줍니다.

AWS CLI

애플리케이션의 음성 채널 상태 및 설정에 대한 정보를 검색하려면

다음 `get-voice-channel` 예제에서는 애플리케이션의 음성 채널 상태 및 설정을 검색합니다.

```
aws pinpoint get-voice-channel \
    --application-id 6e0b7591a90841d2b5d93fa11143e5a7 \
    --region us-east-1
```

출력:

```
{
    "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
    }
}
```

- 자세한 API 내용은 명령 참조GetVoiceChannel의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스의 태그 목록을 검색하려면

다음 `list-tags-for-resource` 예제에서는 지정된 리소스와 연결된 모든 태그(키 이름 및 값)를 검색합니다.

```
aws pinpoint list-tags-for-resource \
    --resource-arn arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example
```

출력:

```
{
    "TagsModel": {
        "tags": {
            "Year": "2019",
            "Stack": "Production"
        }
    }
}
```

자세한 내용은 Amazon Pinpoint 개발자 안내서의 'Amazon Pinpoint 리소스에 태그 지정 <https://docs.aws.amazon.com/pinpoint/latest/developerguide/tagging-resources.html>'__을 참조하세요.

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### `phone-number-validate`

다음 코드 예시에서는 `phone-number-validate`을 사용하는 방법을 보여 줍니다.

AWS CLI

전화번호에 대한 정보를 검색합니다.

다음은 전화번호에 대한 정보를 `phone-number-validate` 검색합니다.

```
aws pinpoint phone-number-validate \
    --number-validate-request PhoneNumber="+12065550142" \
    --region us-east-1
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Pinpoint 사용 설명서의 Amazon Pinpoint SMS 채널을 참조하세요. Amazon Pinpoint

- 자세한 API 내용은 명령 참조PhoneNumberValidate의 섹션을 참조하세요. AWS CLI

### `send-messages`

다음 코드 예시에서는 `send-messages`을 사용하는 방법을 보여 줍니다.

AWS CLI

애플리케이션의 엔드포인트를 사용하여 SMS 메시지를 보내려면

다음 `send-messages` 예시에서는 엔드포인트가 있는 애플리케이션에 다이렉트 메시지를 보냅니다.

```
aws pinpoint send-messages \
    --application-id 611e3e3cdd47474c9c1399a505665b91 \
    --message-request file://myfile.json \
    --region us-west-2
```

`myfile.json`의 콘텐츠:

```
{
    "MessageConfiguration": {
        "SMSMessage": {
            "Body": "hello, how are you?"
```

```
        }
    },
    "Endpoints": {
        "testendpoint": {}
    }
}
```

출력:

```
{
    "MessageResponse": {
        "ApplicationId": "611e3e3cdd47474c9c1399a505665b91",
        "EndpointResult": {
            "testendpoint": {
                "Address": "+12345678900",
                "DeliveryStatus": "SUCCESSFUL",
                "MessageId": "itnuqhai5alf1n6ahv3udc05n7hhddr6gb3lq6g0",
                "StatusCode": 200,
                "StatusMessage": "MessageId:
 itnuqhai5alf1n6ahv3udc05n7hhddr6gb3lq6g0"
            }
        },
        "RequestId": "c7e23264-04b2-4a46-b800-d24923f74753"
    }
}
```

자세한 내용은 Amazon Pinpoint 사용 설명서의 Amazon Pinpoint SMS 채널을 참조하세요. Amazon Pinpoint

- 자세한 API 내용은 명령 참조SendMessages의 섹션을 참조하세요. AWS CLI

### `send-users-messages`

다음 코드 예시에서는 `send-users-messages`을 사용하는 방법을 보여 줍니다.

AWS CLI

애플리케이션 사용자에게 SMS 메시지를 보내려면

다음 `send-users-messages` 예제에서는 애플리케이션 사용자에게 직접 메시지를 보냅니다.

```
aws pinpoint send-users-messages \
```

```
    --application-id 611e3e3cdd47474c9c1399a505665b91 \
    --send-users-message-request file://myfile.json \
    --region us-west-2
```

myfile.json의 콘텐츠:

```
{
    "MessageConfiguration": {
        "SMSMessage": {
            "Body": "hello, how are you?"
        }
    },
    "Users": {
        "testuser": {}
    }
}
```

출력:

```
{
    "SendUsersMessageResponse": {
        "ApplicationId": "611e3e3cdd47474c9c1399a505665b91",
        "RequestId": "e0b12cf5-2359-11e9-bb0b-d5fb91876b25",
        "Result": {
            "testuser": {
                "testuserendpoint": {
                    "DeliveryStatus": "SUCCESSFUL",
                    "MessageId": "7qu4hk5bqhda3i7i2n4pjf98qcuh8b7p45ifsmo0",
                    "StatusCode": 200,
                    "StatusMessage": "MessageId:
 7qu4hk5bqhda3i7i2n4pjf98qcuh8b7p45ifsmo0",
                    "Address": "+12345678900"
                }
            }
        }
    }
}
```

자세한 내용은 Amazon Pinpoint 사용 설명서의 Amazon Pinpoint SMS 채널을 참조하세요.

Amazon Pinpoint

- 자세한 API 내용은 명령 참조SendUsersMessages의 섹션을 참조하세요. AWS CLI

### tag-resource

다음 코드 예시에서는 tag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 태그를 추가하려면

다음 예제에서는 리소스에 두 개의 태그(키 이름 및 값)를 추가합니다.

```
aws pinpoint list-tags-for-resource \
    --resource-arn arn:aws:mobiletargeting:us-
east-1:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example \
    --tags-model tags={Stack=Production,Year=2019}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Pinpoint 개발자 안내서의 'Amazon Pinpoint 리소스에 태그 지정 <https://docs.aws.amazon.com/pinpoint/latest/developerguide/tagging-resources.html>'__을 참조하세요.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### untag-resource

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 리소스에서 태그를 제거하려면

다음 untag-resource 예제에서는 리소스에서 지정된 태그(키 이름 및 값)를 제거합니다.

```
aws pinpoint untag-resource \
    --resource-arn arn:aws:mobiletargeting:us-
west-2:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example \
    --tag-keys Year
```

이 명령은 출력을 생성하지 않습니다.

예제 2: 리소스에서 여러 태그를 제거하려면

다음 untag-resource 예제에서는 리소스에서 지정된 태그(키 이름 및 값)를 제거합니다.

```
aws pinpoint untag-resource \
```

```
    --resource-arn arn:aws:mobiletargeting:us-
east-1:AIDACKCEVSQ6C2EXAMPLE:apps/810c7aab86d42fb2b56c8c966example \
    --tag-keys Year Stack
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Pinpoint 개발자 안내서의 'Amazon Pinpoint 리소스 <https://docs.aws.amazon.com/pinpoint/latest/developerguide/tagging-resources.html> 태그 지정'__을 참조하세요.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-sms-channel

다음 코드 예시에서는 update-sms-channel을 사용하는 방법을 보여 줍니다.

AWS CLI

SMS 채널을 활성화하거나 애플리케이션의 SMS 채널 상태 및 설정을 업데이트합니다.

다음 update-sms-channel 예제에서는 애플리케이션의 SMS 채널에 대한 SMS 채널을 활성화합니다.

```
aws pinpoint update-sms-channel \
    --application-id 611e3e3cdd47474c9c1399a505665b91 \
    --sms-channel-request Enabled=true \
    --region us-west-2
```

출력:

```
{
    "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
```

```
    }
}
```

자세한 내용은 Amazon Pinpoint 사용 설명서의 Amazon Pinpoint SMS 채널을 참조하세요. Amazon Pinpoint

- 자세한 API 내용은 명령 참조UpdateSmsChannel의 섹션을 참조하세요. AWS CLI

## 를 사용한 Amazon Polly 예제 AWS CLI

다음 코드 예제에서는 Amazon Polly 와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `delete-lexicon`

다음 코드 예시에서는 `delete-lexicon`을 사용하는 방법을 보여 줍니다.

AWS CLI

어휘를 삭제하는 방법

다음 `delete-lexicon` 예시에서는 지정된 어휘를 삭제합니다.

```
aws polly delete-lexicon \
    --name w3c
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Polly 개발자 안내서의 DeleteLexicon 작업 사용을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteLexicon의 섹션을 참조하세요. AWS CLI

### get-lexicon

다음 코드 예시에서는 get-lexicon을 사용하는 방법을 보여 줍니다.

AWS CLI

어휘의 콘텐츠를 검색하는 방법

다음 get-lexicon 예시에서는 지정된 발음 어휘의 콘텐츠를 검색합니다.

```
aws polly get-lexicon \
    --name w3c
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 Amazon Polly 개발자 안내서의 GetLexicon 작업 사용을 참조하세요.

- 자세한 API 내용은 명령 참조GetLexicon의 섹션을 참조하세요. AWS CLI

### get-speech-synthesis-task

다음 코드 예시에서는 get-speech-synthesis-task을 사용하는 방법을 보여 줍니다.

AWS CLI

음성 합성 태스크에 대한 정보를 가져오는 방법

다음 get-speech-synthesis-task 예시에서는 지정된 음성 합성 태스크에 대한 정보를 검색합니다.

```
aws polly get-speech-synthesis-task \
    --task-id 70b61c0f-57ce-4715-a247-cae8729dcce9
```

출력:

```
{
    "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"
    }
}
```

자세한 내용을 알아보려면 Amazon Polly 개발자 안내서의 긴 오디오 파일 생성을 참조하세요.

- 자세한 API 내용은 명령 참조GetSpeechSynthesisTask의 섹션을 참조하세요. AWS CLI

### list-lexicons

다음 코드 예시에서는 list-lexicons을 사용하는 방법을 보여 줍니다.

AWS CLI

어휘를 나열하는 방법

다음 list-lexicons 예시에서는 발음 어휘를 나열합니다.

```
aws polly list-lexicons
```

출력:

```
{
    "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
            }
        }
    ]
}
```

자세한 내용은 Amazon Polly 개발자 안내서의 ListLexicons 작업 사용을 참조하세요.

- 자세한 API 내용은 명령 참조ListLexicons의 섹션을 참조하세요. AWS CLI

### list-speech-synthesis-tasks

다음 코드 예시에서는 `list-speech-synthesis-tasks`을 사용하는 방법을 보여 줍니다.

AWS CLI

음성 합성 작업을 나열하려면

다음 `list-speech-synthesis-tasks` 예제에서는 음성 합성 작업을 나열합니다.

```
aws polly list-speech-synthesis-tasks
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용을 알아보려면 Amazon Polly 개발자 안내서의 긴 오디오 파일 생성을 참조하세요.

- 자세한 API 내용은 명령 참조ListSpeechSynthesisTasks의 섹션을 참조하세요. AWS CLI

### put-lexicon

다음 코드 예시에서는 put-lexicon을 사용하는 방법을 보여 줍니다.

AWS CLI

어휘를 저장하는 방법

다음 put-lexicon 예시에서는 지정된 발음 어휘를 저장합니다. example.pls 파일은 W3C PLS 호환 어휘를 지정합니다.

```
aws polly put-lexicon \
    --name w3c \
    --content file://example.pls
```

example.pls의 콘텐츠

```
{
    <?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>
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Polly 개발자 안내서의 PutLexicon 작업 사용을 참조하세요.

- 자세한 API 내용은 명령 참조PutLexicon의 섹션을 참조하세요. AWS CLI

### `start-speech-synthesis-task`

다음 코드 예시에서는 `start-speech-synthesis-task`을 사용하는 방법을 보여 줍니다.

AWS CLI

텍스트를 합성하는 방법

다음 `start-speech-synthesis-task` 예제에서는 에서 텍스트를 합성`text_file.txt`하고 결과 MP3 파일을 지정된 버킷에 저장합니다.

```
aws polly start-speech-synthesis-task \
    --output-format mp3 \
    --output-s3-bucket-name my-s3-bucket \
    --text  file://text_file.txt \
    --voice-id Joanna
```

출력:

```
{
    "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"
    }
}
```

자세한 내용을 알아보려면 Amazon Polly 개발자 안내서의 긴 오디오 파일 생성을 참조하세요.

- 자세한 API 내용은 명령 참조StartSpeechSynthesisTask의 섹션을 참조하세요. AWS CLI

## AWS 가격표 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS 가격표.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `describe-services`

다음 코드 예시에서는 `describe-services`을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스 메타데이터를 검색하려면

이 예제에서는 Amazon EC2 서비스 코드의 메타데이터를 검색합니다.

명령:

```
aws pricing describe-services --service-code AmazonEC2 --format-version aws_v1 --
max-items 1
```

출력:


```
{
  "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"
}
```

- 자세한 API 내용은 명령 참조DescribeServices의 섹션을 참조하세요. AWS CLI

### `get-attribute-values`

다음 코드 예시에서는 `get-attribute-values`을 사용하는 방법을 보여 줍니다.

AWS CLI

속성 값 목록을 검색하려면

다음 `get-attribute-values` 예제에서는 지정된 속성에 사용할 수 있는 값 목록을 검색합니다.

```
aws pricing get-attribute-values \
    --service-code AmazonEC2 \
    --attribute-name volumeType \
    --max-items 2
```

출력:

```
{
    "NextToken": "eyJOZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ==",
    "AttributeValues": [
        {
            "Value": "Cold HDD"
        },
        {
            "Value": "General Purpose"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetAttributeValues의 섹션을 참조하세요. AWS CLI

### get-products

다음 코드 예시에서는 get-products을 사용하는 방법을 보여 줍니다.

AWS CLI

제품 목록을 검색하려면

이 예제에서는 지정된 기준과 일치하는 제품 목록을 검색합니다.

명령:

```
aws pricing get-products --filters file://filters.json --format-version aws_v1 --
max-results 1 --service-code AmazonEC2
```

filter.json:

```
        [
  {
    "Type": "TERM_MATCH",
    "Field": "ServiceCode",
```

```
    "Value": "AmazonEC2"
  },
  {
    "Type": "TERM_MATCH",
    "Field": "volumeType",
    "Value": "Provisioned IOPS"
  }
]
```

출력:

```
{
  "FormatVersion": "aws_v1",
  "NextToken": "WGDY7ko8fQXdlaUZVdasFQ==:RVSagyIFn770XQOzdUIcO9BY6ucBG9itXAZGZF/
zioUzOsUKh6PCcPWaOyPZRiMePb986TeoKYB9l55fw/
CyoMq5ymnGmT1Vj39TljbbAlhcqnVfTmPIilx8Uy5bdDaBYy/e/2Ofw9Edzsykbs8LTBuNbiDQ
+BBds5yeI9AQkUepruKk3aEahFPxJ55kx/zk",
  "PriceList": [
      "{\"product\":{\"productFamily\":\"Storage\",\"attributes\":{\"storageMedia\":
\"SSD-backed\",\"maxThroughputvolume\":\"320 MB/sec\",\"volumeType\":\"Provisioned
 IOPS\",\"maxIopsvolume\":\"20000\",\"servicecode\":\"AmazonEC2\",\"usagetype
\":\"APS1-EBS:VolumeUsage.piops\",\"locationType\":\"AWS Region\",\"location\":
\"Asia Pacific (Singapore)\",\"servicename\":\"Amazon Elastic Compute Cloud\",
\"maxVolumeSize\":\"16 TiB\",\"operation\":\"\"},\"sku\":\"3MKHN58N7RDDVGKJ\"},
\"serviceCode\":\"AmazonEC2\",\"terms\":{\"OnDemand\":{\"3MKHN58N7RDDVGKJ.JRTCKXETXF
\":{\"priceDimensions\":{\"3MKHN58N7RDDVGKJ.JRTCKXETXF.6YS6EN2CT7\":{\"unit\":\"GB-
Mo\",\"endRange\":\"Inf\",\"description\":\"$0.138 per GB-month of Provisioned
 IOPS SSD (io1)  provisioned storage - Asia Pacific (Singapore)\",\"appliesTo
\":[],\"rateCode\":\"3MKHN58N7RDDVGKJ.JRTCKXETXF.6YS6EN2CT7\",\"beginRange\":
\"0\",\"pricePerUnit\":{\"USD\":\"0.1380000000\"}}},\"sku\":\"3MKHN58N7RDDVGKJ
\",\"effectiveDate\":\"2018-08-01T00:00:00Z\",\"offerTermCode\":\"JRTCKXETXF
\",\"termAttributes\":{}}}},\"version\":\"20180808005701\",\"publicationDate\":
\"2018-08-08T00:57:01Z\"}"
  ]
}
```

- 자세한 API 내용은 명령 참조GetProducts의 섹션을 참조하세요. AWS CLI

## AWS Private CA 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS Private CA.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `create-certificate-authority-audit-report`

다음 코드 예시에서는 `create-certificate-authority-audit-report`을 사용하는 방법을 보여 줍니다.

AWS CLI

인증서 기관 감사 보고서를 생성하려면

다음 `create-certificate-authority-audit-report` 명령은 에서 식별한 프라이빗 CA에 대한 감사 보고서를 생성합니다ARN.

```
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
```

- 자세한 API 내용은 명령 참조CreateCertificateAuthorityAuditReport의 섹션을 참조하세요. AWS CLI

### `create-certificate-authority`

다음 코드 예시에서는 `create-certificate-authority`을 사용하는 방법을 보여 줍니다.

AWS CLI

프라이빗 인증 기관을 생성하려면

다음 create-certificate-authority 명령은 AWS 계정에 프라이빗 인증 기관을 생성합니다.

```
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
```

- 자세한 API 내용은 명령 참조CreateCertificateAuthority의 섹션을 참조하세요. AWS CLI

### delete-certificate-authority

다음 코드 예시에서는 delete-certificate-authority을 사용하는 방법을 보여 줍니다.

AWS CLI

프라이빗 인증 기관을 삭제하려면

다음 delete-certificate-authority 명령은 에서 식별한 인증서 기관을 삭제합니다ARN.

```
aws acm-pca delete-certificate-authority --certificate-
authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-
authority/12345678-1234-1234-1234-123456789012
```

- 자세한 API 내용은 명령 참조DeleteCertificateAuthority의 섹션을 참조하세요. AWS CLI

### describe-certificate-authority-audit-report

다음 코드 예시에서는 describe-certificate-authority-audit-report을 사용하는 방법을 보여 줍니다.

AWS CLI

인증 기관에 대한 감사 보고서를 설명하려면

다음 describe-certificate-authority-audit-report 명령은 에 의해 식별된 CA에 대해 지정된 감사 보고서에 대한 정보를 나열합니다ARN.

```
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
```

- 자세한 API 내용은 명령 참조DescribeCertificateAuthorityAuditReport의 섹션을 참조하세요. AWS CLI

### describe-certificate-authority

다음 코드 예시에서는 describe-certificate-authority을 사용하는 방법을 보여 줍니다.

AWS CLI

프라이빗 인증 기관을 설명하려면

다음 describe-certificate-authority 명령은 에서 식별한 프라이빗 CA에 대한 정보를 나열합니다ARN.

```
aws acm-pca describe-certificate-authority --certificate-
authority-arn arn:aws:acm-pca:us-west-2:123456789012:certificate-
authority/12345678-1234-1234-1234-123456789012
```

- 자세한 API 내용은 명령 참조DescribeCertificateAuthority의 섹션을 참조하세요. AWS CLI

### get-certificate-authority-certificate

다음 코드 예시에서는 get-certificate-authority-certificate을 사용하는 방법을 보여 줍니다.

AWS CLI

CA(인증 기관) 인증서를 검색하려면

다음 get-certificate-authority-certificate 명령은 에서 지정한 프라이빗 CA의 인증서 및 인증서 체인을 검색합니다ARN.

```
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
```

- 자세한 API 내용은 명령 참조GetCertificateAuthorityCertificate의 섹션을 참조하세요. AWS CLI

### get-certificate-authority-csr

다음 코드 예시에서는 get-certificate-authority-csr을 사용하는 방법을 보여 줍니다.

AWS CLI

인증서 기관에 대한 인증서 서명 요청을 검색하려면

다음 `get-certificate-authority-csr` 명령은 에서 지정한 프라이빗 CA에 CSR 대해 를 검색합니다ARN.

```
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
```

- 자세한 API 내용은 명령 참조GetCertificateAuthorityCsr의 섹션을 참조하세요. AWS CLI

## `get-certificate`

다음 코드 예시에서는 `get-certificate`을 사용하는 방법을 보여 줍니다.

AWS CLI

발급된 인증서를 검색하려면

다음 `get-certificate` 예제에서는 지정된 프라이빗 CA에서 인증서를 검색합니다.

```
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
```

출력:

```
-----BEGIN CERTIFICATE-----
MIIEDzCCAvegAwIBAgIRAJuJ8f6ZVYL7gG/rS3qvrZMwDQYJKoZIhvcNAQELBQAw
cTELMAkGA1UEBhMCVVMxEzARBgNVBAgMCldhc2hpbmd0b24xEDAOBgNVBAcMB1Nl
    ....certificate body truncated for brevity....
tKCSglgZZrd4FdLw1EkGm+UVXnodwMtJEQyy3oTfZjURPIyyaqskTu/KSS7YDjK0
KQNy73D6LtmdOEbAyq10XiDxqY41lvKHJ1eZrPaBmYNABxU=
-----END CERTIFICATE---- -----BEGIN CERTIFICATE-----
MIIDrzCCApegAwIBAgIRAOskdzLvcj1eShkoyEE693AwDQYJKoZIhvcNAQELBQAw
```

```
cTELMAkGA1UEBhMCVVMxEzARBgNVBAgMCldhc2hpbmd0b24xEDAOBgNVBAcMB1Nl
    ...certificate body truncated for brevity....
kdRGB6P2hpxstDOUIwAoCbhoaWwfA4ybJznf+jOQhAziNlRdKQRR8nODWpKt7H9w
dJ5nxsTk/fniJz86Ddtp6n8s82wYdkN3cVffeK72A9aTCOU=
-----END CERTIFICATE-----
```

출력의 첫 번째 부분은 인증서 자체입니다. 두 번째 부분은 루트 CA 인증서에 연결하는 인증서 체인입니다. `--output text` 옵션을 사용하면 두 인증서 조각 사이에 TAB 문자가 삽입됩니다(인덴트 텍스트의 원인). 이 출력을 가져와 다른 도구로 인증서를 구문 분석하려는 경우 올바르게 처리되도록 TAB 문자를 제거해야 할 수 있습니다.

- 자세한 API 내용은 명령 참조GetCertificate의 섹션을 참조하세요. AWS CLI

### `import-certificate-authority-certificate`

다음 코드 예시에서는 `import-certificate-authority-certificate`을 사용하는 방법을 보여 줍니다.

AWS CLI

인증서 기관 인증서를 로 가져오려면 ACM PCA

다음 `import-certificate-authority-certificate` 명령은 에서 지정한 CA에 대해 서명된 프라이빗 CA 인증서를 ACM ARN로 가져옵니다PCA.

```
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
```

- 자세한 API 내용은 명령 참조ImportCertificateAuthorityCertificate의 섹션을 참조하세요. AWS CLI

### `issue-certificate`

다음 코드 예시에서는 `issue-certificate`을 사용하는 방법을 보여 줍니다.

AWS CLI

프라이빗 인증서를 발급하려면

다음 issue-certificate 명령은 에서 지정한 프라이빗 CAARN를 사용하여 프라이빗 인증서를 발급합니다.

```
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
```

- 자세한 API 내용은 명령 참조IssueCertificate의 섹션을 참조하세요. AWS CLI

### list-certificate-authorities

다음 코드 예시에서는 list-certificate-authorities을 사용하는 방법을 보여 줍니다.

AWS CLI

프라이빗 인증 기관을 나열하려면

다음 list-certificate-authorities 명령은 계정의 모든 프라이빗CAs에 대한 정보를 나열합니다.

```
aws acm-pca list-certificate-authorities --max-results 10
```

- 자세한 API 내용은 명령 참조ListCertificateAuthorities의 섹션을 참조하세요. AWS CLI

### list-tags

다음 코드 예시에서는 list-tags을 사용하는 방법을 보여 줍니다.

AWS CLI

인증 기관의 태그를 나열하려면

다음 list-tags 명령은 에서 지정한 프라이빗 CA와 연결된 태그를 나열합니다ARN.

```
aws acm-pca list-tags --certificate-authority-arn arn:aws:acm-pca:us-
west-2:123456789012:certificate-authority/123455678-1234-1234-1234-123456789012 --
max-results 10
```

- 자세한 API 내용은 명령 참조ListTags의 섹션을 참조하세요. AWS CLI

### revoke-certificate

다음 코드 예시에서는 revoke-certificate을 사용하는 방법을 보여 줍니다.

AWS CLI

프라이빗 인증서를 취소하려면

다음 revoke-certificate 명령은 에서 식별한 CA에서 프라이빗 인증서를 취소합니다ARN.

```
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"
```

- 자세한 API 내용은 명령 참조RevokeCertificate의 섹션을 참조하세요. AWS CLI

### tag-certificate-authority

다음 코드 예시에서는 tag-certificate-authority을 사용하는 방법을 보여 줍니다.

AWS CLI

프라이빗 인증 기관에 태그를 연결하려면

다음 tag-certificate-authority 명령은 프라이빗 CA에 하나 이상의 태그를 연결합니다.

```
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
```

- 자세한 API 내용은 명령 참조TagCertificateAuthority의 섹션을 참조하세요. AWS CLI

### untag-certificate-authority

다음 코드 예시에서는 untag-certificate-authority을 사용하는 방법을 보여 줍니다.

AWS CLI

프라이빗 인증 기관에서 하나 이상의 태그를 제거하려면

다음 `untag-certificate-authority` 명령은 에서 식별한 프라이빗 CA에서 태그를 제거합니다ARN.

```
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
```

- 자세한 API 내용은 명령 참조UntagCertificateAuthority의 섹션을 참조하세요. AWS CLI

**`update-certificate-authority`**

다음 코드 예시에서는 `update-certificate-authority`을 사용하는 방법을 보여 줍니다.

AWS CLI

프라이빗 인증 기관의 구성을 업데이트하려면

다음 `update-certificate-authority` 명령은 에서 식별한 프라이빗 CA의 상태 및 구성을 업데이트합니다ARN.

```
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"
```

- 자세한 API 내용은 명령 참조UpdateCertificateAuthority의 섹션을 참조하세요. AWS CLI

## AWS Proton 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS Proton.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### cancel-service-instance-deployment

다음 코드 예시에서는 cancel-service-instance-deployment을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스 인스턴스 배포를 취소하려면

다음 cancel-service-instance-deployment 예제에서는 서비스 인스턴스 배포를 취소합니다.

```
aws proton cancel-service-instance-deployment \
    --service-instance-name "instance-one" \
    --service-name "simple-svc"
```

출력:

```
{
    "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"
```

```
    }
}
```

자세한 내용은 AWS Proton 관리자 안내서의 서비스 인스턴스 업데이트 또는 AWS Proton 사용 설명서의 서비스 인스턴스 업데이트를 참조하세요.

- 자세한 API 내용은 명령 참조CancelServiceInstanceDeployment의 섹션을 참조하세요. AWS CLI

### `cancel-service-pipeline-deployment`

다음 코드 예시에서는 `cancel-service-pipeline-deployment`을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스 파이프라인 배포를 취소하려면

다음 `cancel-service-pipeline-deployment` 예제에서는 서비스 파이프라인 배포를 취소합니다.

```
aws proton cancel-service-pipeline-deployment \
    --service-name "simple-svc"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS Proton 관리자 안내서의 서비스 파이프라인 업데이트 또는 AWS Proton 사용 설명서의 서비스 파이프라인 업데이트를 참조하세요.

- 자세한 API 내용은 명령 참조CancelServicePipelineDeployment의 섹션을 참조하세요. AWS CLI

### create-service

다음 코드 예시에서는 create-service을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스를 생성하려면

다음 create-service 예제에서는 서비스 파이프라인을 사용하여 서비스를 생성합니다.

```
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
```

spec.yaml의 콘텐츠:

```
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"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS Proton 관리자 안내서의 서비스 생성 및 AWS Proton 사용 설명서의 서비스 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateService의 섹션을 참조하세요. AWS CLI

### `delete-service`

다음 코드 예시에서는 `delete-service`을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스를 삭제하는 방법

다음 `delete-service` 예제에서는 서비스를 삭제합니다.

```
aws proton delete-service \
    --name "simple-svc"
```

출력:

```
{
    "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"
```

```
    }
}
```

자세한 내용은  AWS Proton 관리자 안내서의 서비스 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteService의 섹션을 참조하세요. AWS CLI

### get-service-instance

다음 코드 예시에서는 get-service-instance을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스 인스턴스 세부 정보를 가져오려면

다음 get-service-instance 예제에서는 서비스 인스턴스에 대한 세부 데이터를 가져옵니다.

```
aws proton get-service-instance \
    --name "instance-one" \
    --service-name "simple-svc"
```

출력:

```
{
    "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"
```

```
    }
}
```

자세한 내용은 AWS Proton 관리자 안내서의 서비스 데이터 보기 또는 AWS Proton 사용 설명서의 서비스 데이터 보기를 참조하세요.

- 자세한 API 내용은 명령 참조GetServiceInstance의 섹션을 참조하세요. AWS CLI

### get-service

다음 코드 예시에서는 get-service을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스 세부 정보를 가져오려면

다음 get-service 예제에서는 서비스에 대한 세부 데이터를 가져옵니다.

```
aws proton get-service \
    --name "simple-svc"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS Proton 관리자 안내서의 서비스 데이터 보기 또는 AWS Proton 사용 설명서의 서비스 데이터 보기를 참조하세요.

- 자세한 API 내용은 명령 참조GetService의 섹션을 참조하세요. AWS CLI

### `list-service-instances`

다음 코드 예시에서는 `list-service-instances`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 모든 서비스 인스턴스 나열

다음 `list-service-instances` 예제에서는 서비스 인스턴스를 나열합니다.

```
aws proton list-service-instances
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS Proton 관리자 안내서의 서비스 인스턴스 데이터 보기 또는 AWS Proton 사용 설명서의 서비스 인스턴스 데이터 보기를 참조하세요.

예제 2: 지정된 서비스 인스턴스를 나열하려면

다음 `get-service-instance` 예제에서는 서비스 인스턴스를 가져옵니다.

```
aws proton get-service-instance \
    --name "instance-one" \
    --service-name "simple-svc"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS Proton 관리자 안내서의 서비스 인스턴스 데이터 보기 또는 AWS Proton 사용 설명서의 서비스 인스턴스 데이터 보기를 참조하세요.

- 자세한 API 내용은 명령 참조ListServiceInstances의 섹션을 참조하세요. AWS CLI

### update-service-instance

다음 코드 예시에서는 update-service-instance을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스 인스턴스를 새 마이너 버전으로 업데이트하려면

다음 update-service-instance 예제에서는 서비스 인스턴스를 서비스 템플릿의 새 마이너 버전으로 업데이트하여 'my-other-instance'라는 새 인스턴스를 필요한 새 입력과 함께 추가합니다.

```
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"
```

service-spec.yaml의 콘텐츠:

```
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"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은  AWS Proton 관리자 안내서의 서비스 인스턴스 업데이트 또는  AWS Proton 사용 설명서의 서비스 인스턴스 업데이트를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateServiceInstance의 섹션을 참조하세요. AWS CLI

### update-service-pipeline

다음 코드 예시에서는 update-service-pipeline을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스 파이프라인을 업데이트하려면

다음 update-service-pipeline 예제에서는 서비스 파이프라인을 서비스 템플릿의 새 마이너 버전으로 업데이트합니다.

```
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"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은  AWS Proton 관리자 안내서의 서비스 파이프라인 업데이트 또는  AWS Proton 사용 설명서의 서비스 파이프라인 업데이트를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateServicePipeline의 섹션을 참조하세요. AWS CLI

### update-service

다음 코드 예시에서는 update-service을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스를 업데이트하려면

다음 update-service 예제에서는 서비스 설명을 편집합니다.

```
aws proton update-service \
    --name "MySimpleService" \
    --description "Edit by updating description"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은  AWS Proton 관리자 안내서의 서비스 편집 또는  AWS Proton 사용 설명서의 서비스 편집을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateService의 섹션을 참조하세요. AWS CLI

## QLDB 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다QLDB.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### `cancel-journal-kinesis-stream`

다음 코드 예시에서는 `cancel-journal-kinesis-stream`을 사용하는 방법을 보여 줍니다.

AWS CLI

저널 스트림을 취소하려면

다음 `cancel-journal-kinesis-stream` 예제는 원장에서 지정된 저널 스트림을 취소합니다.

```
aws qldb cancel-journal-kinesis-stream \
    --ledger-name myExampleLedger \
    --stream-id 7ISCkqwe4y25YyHLzYUFAf
```

출력:

```
{
    "StreamId": "7ISCkqwe4y25YyHLzYUFAf"
}
```

자세한 내용은 Amazon 개발자 안내서의 Amazon에서 저널 데이터 스트리밍QLDB을 참조하세요. QLDB

- 자세한 API 내용은 명령 참조CancelJournalKinesisStream의 섹션을 참조하세요. AWS CLI

### `create-ledger`

다음 코드 예시에서는 `create-ledger`을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: 기본 속성을 사용하여 원장을 생성하는 방법

다음 `create-ledger` 예시에서는 `myExampleLedger` 이름과 `STANDARD` 권한 모드를 사용하여 원장을 생성합니다. 삭제 방지 및 AWS KMS 키에 대한 선택적 파라미터는 지정되지 않으므로 각각 `true` 및 AWS 소유 KMS 키로 기본 설정됩니다.

```
aws qldb create-ledger \
    --name myExampleLedger \
    --permissions-mode STANDARD
```

출력:

```
{
    "State": "CREATING",
    "Arn": "arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger",
    "DeletionProtection": true,
    "CreationDateTime": 1568839243.951,
    "Name": "myExampleLedger",
    "PermissionsMode": "STANDARD"
}
```

예제 2: 삭제 방지가 비활성화된 원장, 고객 관리형 KMS 키 및 지정된 태그를 생성하려면

다음 `create-ledger` 예시에서는 `myExampleLedger2` 이름과 `STANDARD` 권한 모드를 사용하여 원장을 생성합니다. 삭제 보호 기능이 비활성화되고 지정된 고객 관리형 KMS 키가 저장 시 암호화에 사용되며 지정된 태그가 리소스에 연결됩니다.

```
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
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon 개발자 안내서의 Amazon QLDB Ledgers 기본 작업을 참조하세요. QLDB

- 자세한 API 내용은 명령 참조CreateLedger의 섹션을 참조하세요. AWS CLI

### delete-ledger

다음 코드 예시에서는 delete-ledger을 사용하는 방법을 보여 줍니다.

AWS CLI

원장을 삭제하려면

다음 delete-ledger 예제에서는 지정된 원장을 삭제합니다.

```
aws qldb delete-ledger \
    --name myExampleLedger
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon 개발자 안내서의 Amazon QLDB Ledgers 기본 작업을 참조하세요. QLDB

- 자세한 API 내용은 명령 참조DeleteLedger의 섹션을 참조하세요. AWS CLI

### describe-journal-kinesis-stream

다음 코드 예시에서는 describe-journal-kinesis-stream을 사용하는 방법을 보여 줍니다.

AWS CLI

저널 스트림을 설명하려면

다음 describe-journal-kinesis-stream 예제에서는 원장의 지정된 저널 스트림에 대한 세부 정보를 표시합니다.

```
aws qldb describe-journal-kinesis-stream \
    --ledger-name myExampleLedger \
    --stream-id 7ISCkqwe4y25YyHLzYUFAf
```

출력:

```
{
```

```
    "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"
    }
}
```

자세한 내용은 Amazon 개발자 안내서의 Amazon에서 저널 데이터 스트리밍QLDB을 참조하세요. QLDB

- 자세한 API 내용은 명령 참조DescribeJournalKinesisStream의 섹션을 참조하세요. AWS CLI

### describe-journal-s3-export

다음 코드 예시에서는 describe-journal-s3-export을 사용하는 방법을 보여 줍니다.

AWS CLI

저널 내보내기 작업을 설명하려면

다음 describe-journal-s3-export 예제에서는 원장의 지정된 내보내기 작업에 대한 세부 정보를 표시합니다.

```
aws qldb describe-journal-s3-export \
    --name myExampleLedger \
    --export-id ADR2ONPKN5LINYGb4dp7yZ
```

출력:

```
{
```

```
    "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": "ADR2ONPKN5LINYGb4dp7yZ"
    }
}
```

자세한 내용은 Amazon 개발자 안내서의 Amazon에서 저널 내보내기QLDB를 참조하세요. QLDB

- API 자세한 내용은 AWS CLI 명령 참조의 DescribeJournalS3Export를 참조하세요.

### describe-ledger

다음 코드 예시에서는 describe-ledger을 사용하는 방법을 보여 줍니다.

AWS CLI

원장을 설명하려면

다음 describe-ledger 예제에서는 지정된 원장에 대한 세부 정보를 표시합니다.

```
aws qldb describe-ledger \
    --name myExampleLedger
```

출력:

```
{
    "CreationDateTime": 1568839243.951,
    "Arn": "arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger",
    "State": "ACTIVE",
    "Name": "myExampleLedger",
```

```
    "DeletionProtection": true,
    "PermissionsMode": "STANDARD",
    "EncryptionDescription": {
        "KmsKeyArn": "arn:aws:kms:us-west-2:123456789012:key/a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111",
        "EncryptionStatus": "ENABLED"
    }
}
```

자세한 내용은 Amazon 개발자 안내서의 Amazon QLDB Ledgers 기본 작업을 참조하세요. QLDB

- 자세한 API 내용은 명령 참조DescribeLedger의 섹션을 참조하세요. AWS CLI

### export-journal-to-s3

다음 코드 예시에서는 export-journal-to-s3을 사용하는 방법을 보여 줍니다.

AWS CLI

저널 블록을 S3로 내보내려면

다음 export-journal-to-s3 예제에서는 이름이 인 원장의 지정된 날짜 및 시간 범위 내에 저널 블록에 대한 내보내기 작업을 생성합니다myExampleLedger. 내보내기 작업은 지정된 Amazon S3 버킷에 블록을 씁니다.

```
aws qldb export-journal-to-s3 \
    --name myExampleLedger \
    --inclusive-start-time 2019-09-18T00:00:00Z \
    --exclusive-end-time 2019-09-18T22:59:59Z \
    --role-arn arn:aws:iam::123456789012:role/my-s3-export-role \
    --s3-export-configuration file://my-s3-export-config.json
```

my-s3-export-config.json의 콘텐츠:

```
{
    "Bucket": "awsExampleBucket",
    "Prefix": "ledgerexport1/",
    "EncryptionConfiguration": {
        "ObjectEncryptionType": "SSE_S3"
    }
}
```

출력:

```
{
    "ExportId": "ADR2ONPKN5LINYGb4dp7yZ"
}
```

자세한 내용은 Amazon 개발자 안내서의 Amazon에서 저널 내보내기QLDB를 참조하세요. QLDB

- API 자세한 내용은 AWS CLI 명령 참조의 ExportJournalToS3를 참조하세요.

### get-block

다음 코드 예시에서는 get-block을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 입력 파일을 사용하여 검증할 저널 블록 및 증거를 가져오는 방법

다음 get-block 예제에서는 블록 데이터 객체와 지정된 원장의 증거를 요청합니다. 요청은 지정된 다이제스트 팁 주소 및 블록 주소에 대한 것입니다.

```
aws qldb get-block \
    --name vehicle-registration \
    --block-address file://myblockaddress.json \
    --digest-tip-address file://mydigesttipaddress.json
```

myblockaddress.json의 콘텐츠:

```
{
    "IonText": "{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl\",sequenceNo:100}"
}
```

mydigesttipaddress.json의 콘텐츠:

```
{
    "IonText": "{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl\",sequenceNo:123}"
}
```

출력:

```
{
    "Block": {
        "IonText": "{blockAddress:{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl
\",sequenceNo:100},transactionId:\"FnQeJBAicTX0Ah32ZnVtSX
\",blockTimestamp:2019-09-16T19:37:05.360Z,blockHash:
{{NoChM92yKRuJAb/jeLd1VnYn4DHiWIf071ACfic9uHc=}},entriesHash:
{{l05LOsiKV14SDbuaYnH7uwXzUvqzIwUiRLXGbTyj/nY=}},previousBlockHash:
{{7kewBXhpdbClcZKxhVmpoMHpUGOJtWQD0iY2LPfZkYA=}},entriesHashList:
[{{eRSwnmAM7WWANWDd5iGOyK+T4tDXyzUq6HZ/0fgLHos=}},{{mHVex/
yjHAWjFPpwhBuH2GKXmKJjK2FBa9faqoUVNtg=}},
{{y5cCBr7pOAIUfsVQ1j0TqtE97b4b4oo1R0vnYyE5wWM=}},{{TvTXygML1bMe6NvEZtGkX
+KR+W/EJl4qD1mmV77KZQg=}}],transactionInfo:{statements:[{statement:
\"FROM VehicleRegistration AS r \\nWHERE r.VIN = '1N4AL11D75C109151'\
\nINSERT INTO r.Owners.SecondaryOwners\\n    VALUE { 'PersonId' :
 'CMVdR77XP8zAglmmFDGTvt' }\",startTime:2019-09-16T19:37:05.302Z,statementDigest:
{{jcgPX2vsOJ0waum4qmDYtn1pCAT9xKNIzA+2k4R+mxA=}}}],documents:
{JUJgkIcNbhS2goq8RqLuZ4:{tableName:\"VehicleRegistration\",tableId:
\"BFJKdXgzt9oF4wjMbuxy4G\",statements:[0]}}},revisions:[{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:\"CMVdR77XP8zAglmmFDGTvt\"}]},City:\"Everett\"},metadata:{id:
\"JUJgkIcNbhS2goq8RqLuZ4\",version:3,txTime:2019-09-16T19:37:05.344Z,txId:
\"FnQeJBAicTX0Ah32ZnVtSX\"}}]}"
    },
    "Proof": {
        "IonText": "[{{l3+EXs69K1+rehlqyWLkt+oHDlw4Zi9pCLW/t/mgTPM=}},
{{48CXG3ehPqsxCYd34EEa8Fso0ORpWWAO8010RJKf3Do=}},{{9UnwnKSQT0i3ge1JMVa
+tMIqCEDaOPTkWxmyHSn8UPQ=}},{{3nW6Vryghk+7pd6wFCtLufgPM6qXHyTNeCb1sCwcDaI=}},
{{Irb5fNhBrNEQ1VPhzlnGT/ZQPadSmgfdtMYcwkNOxoI=}},{{+3CWpYG/ytf/
vq9GidpzSx6JJiLXt1hMQWNnqOy3jfY=}},{{NPx6cRhwsiy5m9UEWS5JTJrZoUdO2jBOAAOmyZAT
+qE=}}]"
    }
}
```

자세한 내용은 Amazon 개발자 안내서의 Amazon에서 데이터 확인을 QLDB 참조하세요. QLDB

예제 2: 짧은 구문을 사용하여 저널 블록 및 검증 증명을 가져오려면

다음 get-block 예제에서는 짧은 구문을 사용하여 블록 데이터 객체와 지정된 원장의 증거를 요청합니다. 요청은 지정된 다이제스트 팁 주소 및 블록 주소에 대한 것입니다.

```
aws qldb get-block \
    --name vehicle-registration \
    --block-address 'IonText="{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl\",sequenceNo:100}"'
 \
    --digest-tip-address 'IonText="{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl
\",sequenceNo:123}"'
```

출력:

```
{
    "Block": {
        "IonText": "{blockAddress:{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl
\",sequenceNo:100},transactionId:\"FnQeJBAicTX0Ah32ZnVtSX
\",blockTimestamp:2019-09-16T19:37:05.360Z,blockHash:
{{NoChM92yKRuJAb/jeLd1VnYn4DHiWIf071ACfic9uHc=}},entriesHash:
{{l05LOsiKV14SDbuaYnH7uwXzUvqzIwUiRLXGbTyj/nY=}},previousBlockHash:
{{7kewBXhpdbClcZKxhVmpoMHpUGOJtWQD0iY2LPfZkYA=}},entriesHashList:
[{{eRSwnmAM7WWANWDd5iGOyK+T4tDXyzUq6HZ/0fgLHos=}},{{mHVex/
yjHAWjFPpwhBuH2GKXmKJjK2FBa9faqoUVNtg=}},
{{y5cCBr7pOAIUfsVQ1j0TqtE97b4b4oo1R0vnYyE5wWM=}},{{TvTXygML1bMe6NvEZtGkX
+KR+W/EJl4qD1mmV77KZQg=}}],transactionInfo:{statements:[{statement:
\"FROM VehicleRegistration AS r \\nWHERE r.VIN = '1N4AL11D75C109151'\
\nINSERT INTO r.Owners.SecondaryOwners\\n    VALUE { 'PersonId' :
 'CMVdR77XP8zAglmmFDGTvt' }\",startTime:2019-09-16T19:37:05.302Z,statementDigest:
{{jcgPX2vsOJ0waum4qmDYtn1pCAT9xKNIzA+2k4R+mxA=}}}],documents:
{JUJgkIcNbhS2goq8RqLuZ4:{tableName:\"VehicleRegistration\",tableId:
\"BFJKdXgzt9oF4wjMbuxy4G\",statements:[0]}}},revisions:[{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:\"CMVdR77XP8zAglmmFDGTvt\"}]},City:\"Everett\"},metadata:{id:
\"JUJgkIcNbhS2goq8RqLuZ4\",version:3,txTime:2019-09-16T19:37:05.344Z,txId:
\"FnQeJBAicTX0Ah32ZnVtSX\"}}]}"
    },
    "Proof": {
        "IonText": "[{{l3+EXs69K1+rehlqyWLkt+oHDlw4Zi9pCLW/t/mgTPM=}},
{{48CXG3ehPqsxCYd34EEa8Fso0ORpWWAO8010RJKf3Do=}},{{9UnwnKSQT0i3ge1JMVa
+tMIqCEDaOPTkWxmyHSn8UPQ=}},{{3nW6Vryghk+7pd6wFCtLufgPM6qXHyTNeCb1sCwcDaI=}},
{{Irb5fNhBrNEQ1VPhzlnGT/ZQPadSmgfdtMYcwkNOxoI=}},{{+3CWpYG/ytf/
vq9GidpzSx6JJiLXt1hMQWNnqOy3jfY=}},{{NPx6cRhwsiy5m9UEWS5JTJrZoUdO2jBOAAOmyZAT
+qE=}}]"
```

```
    }
}
```

자세한 내용은 Amazon 개발자 안내서의 Amazon에서 데이터 확인을 QLDB 참조하세요. QLDB

- 자세한 API 내용은 명령 참조GetBlock의 섹션을 참조하세요. AWS CLI

### get-digest

다음 코드 예시에서는 get-digest을 사용하는 방법을 보여 줍니다.

AWS CLI

원장의 다이제스트를 가져오려면

다음 get-digest 예제는 저널의 가장 최근 커밋 블록에서 지정된 원장으로부터 다이제스트를 요청합니다.

```
aws qldb get-digest \
    --name vehicle-registration
```

출력:

```
{
    "Digest": "6m6BMXobbJKpMhahwVthAEsN6awgnHK62Qq5McGP1Gk=",
    "DigestTipAddress": {
        "IonText": "{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl\",sequenceNo:123}"
    }
}
```

자세한 내용은 Amazon 개발자 안내서의 Amazon에서 데이터 확인을 QLDB 참조하세요. QLDB

- 자세한 API 내용은 명령 참조GetDigest의 섹션을 참조하세요. AWS CLI

### get-revision

다음 코드 예시에서는 get-revision을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 입력 파일을 사용하여 문서 개정 및 검증 증명을 가져오려면

다음 get-revision 예제에서는 수정 데이터 객체와 지정된 원장의 증거를 요청합니다. 이 요청은 지정된 다이제스트 팁 주소, 문서 ID 및 개정의 블록 주소에 대한 것입니다.

```
aws qldb get-revision \
    --name vehicle-registration \
    --block-address file://myblockaddress.json \
    --document-id JUJgkIcNbhS2goq8RqLuZ4 \
    --digest-tip-address file://mydigesttipaddress.json
```

myblockaddress.json의 콘텐츠:

```
{
    "IonText": "{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl\",sequenceNo:100}"
}
```

mydigesttipaddress.json의 콘텐츠:

```
{
    "IonText": "{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl\",sequenceNo:123}"
}
```

출력:

```
{
    "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:\"CMVdR77XP8zAglmmFDGTvt\"}]},City:\"Everett\"},metadata:{id:
\"JUJgkIcNbhS2goq8RqLuZ4\",version:3,txTime:2019-09-16T19:37:05.344Z,txId:
\"FnQeJBAicTX0Ah32ZnVtSX\"}}"
    },
    "Proof": {
        "IonText": "[{{eRSwnmAM7WWANWDd5iGOyK+T4tDXyzUq6HZ/0fgLHos=}},{{VV1rdaNuf
+yJZVGlmsM6gr2T52QvBO8Lg+KgpjcnWAU=}},
{{7kewBXhpdbClcZKxhVmpoMHpUGOJtWQD0iY2LPfZkYA=}},{{l3+EXs69K1+rehlqyWLkt
+oHDlw4Zi9pCLW/t/mgTPM=}},{{48CXG3ehPqsxCYd34EEa8Fso0ORpWWAO8010RJKf3Do=}},
{{9UnwnKSQT0i3ge1JMVa+tMIqCEDaOPTkWxmyHSn8UPQ=}},{{3nW6Vryghk
+7pd6wFCtLufgPM6qXHyTNeCb1sCwcDaI=}},{{Irb5fNhBrNEQ1VPhzlnGT/
```

```
ZQPadSmgfdtMYcwkNOxoI=}},{{+3CWpYG/ytf/vq9GidpzSx6JJiLXt1hMQWNnqOy3jfY=}},
{{NPx6cRhwsiy5m9UEWS5JTJrZoUdO2jBOAAOmyZAT+qE=}}]"
    }
}
```

자세한 내용은 Amazon 개발자 안내서의 Amazon에서 데이터 확인을 QLDB 참조하세요. QLDB

예제 2: 단축 구문을 사용하여 문서 개정 및 확인 증거를 가져오려면

다음 `get-revision` 예제에서는 짧은 구문을 사용하여 수정 데이터 객체와 지정된 원장의 증거를 요청합니다. 이 요청은 지정된 다이제스트 팁 주소, 문서 ID 및 개정의 블록 주소에 대한 것입니다.

```
aws qldb get-revision \
    --name vehicle-registration \
    --block-address 'IonText="{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl\",sequenceNo:100}"'
 \
    --document-id JUJgkIcNbhS2goq8RqLuZ4 \
    --digest-tip-address 'IonText="{strandId:\"KmA3ZZca7vAIiJAK9S5Iwl
\",sequenceNo:123}"'
```

출력:

```
{
    "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:\"CMVdR77XP8zAglmmFDGTvt\"}]},City:\"Everett\"},metadata:{id:
\"JUJgkIcNbhS2goq8RqLuZ4\",version:3,txTime:2019-09-16T19:37:05.344Z,txId:
\"FnQeJBAicTX0Ah32ZnVtSX\"}}"
    },
    "Proof": {
        "IonText": "[{{eRSwnmAM7WWANWDd5iGOyK+T4tDXyzUq6HZ/0fgLHos=}},{{VV1rdaNuf
+yJZVGlmsM6gr2T52QvBO8Lg+KgpjcnWAU=}},
{{7kewBXhpdbClcZKxhVmpoMHpUGOJtWQD0iY2LPfZkYA=}},{{l3+EXs69K1+rehlqyWLkt
+oHDlw4Zi9pCLW/t/mgTPM=}},{{48CXG3ehPqsxCYd34EEa8Fso0ORpWWAO8010RJKf3Do=}},
{{9UnwnKSQT0i3ge1JMVa+tMIqCEDaOPTkWxmyHSn8UPQ=}},{{3nW6Vryghk
+7pd6wFCtLufgPM6qXHyTNeCb1sCwcDaI=}},{{Irb5fNhBrNEQ1VPhzlnGT/
ZQPadSmgfdtMYcwkNOxoI=}},{{+3CWpYG/ytf/vq9GidpzSx6JJiLXt1hMQWNnqOy3jfY=}},
{{NPx6cRhwsiy5m9UEWS5JTJrZoUdO2jBOAAOmyZAT+qE=}}]"
    }
```

```
}
```

자세한 내용은 Amazon 개발자 안내서의 Amazon에서 데이터 확인을 QLDB 참조하세요.  QLDB

- 자세한 API 내용은 명령 참조GetRevision의 섹션을 참조하세요. AWS CLI

### list-journal-kinesis-streams-for-ledger

다음 코드 예시에서는 list-journal-kinesis-streams-for-ledger을 사용하는 방법을 보여 줍니다.

AWS CLI

원장의 저널 스트림을 나열하려면

다음 list-journal-kinesis-streams-for-ledger 예제에서는 지정된 원장의 저널 스트림을 나열합니다.

```
aws qldb list-journal-kinesis-streams-for-ledger \
    --ledger-name myExampleLedger
```

출력:

```
{
    "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"
        }
    ]
```

```
}
```

자세한 내용은 Amazon 개발자 안내서의 Amazon에서 저널 데이터 스트리밍QLDB을 참조하세요. QLDB

- 자세한 API 내용은 명령 참조ListJournalKinesisStreamsForLedger의 섹션을 참조하세요. AWS CLI

### `list-journal-s3-exports-for-ledger`

다음 코드 예시에서는 `list-journal-s3-exports-for-ledger`을 사용하는 방법을 보여 줍니다.

AWS CLI

원장의 저널 내보내기 작업을 나열하려면

다음 `list-journal-s3-exports-for-ledger` 예제에서는 지정된 원장의 저널 내보내기 작업을 나열합니다.

```
aws qldb list-journal-s3-exports-for-ledger \
    --name myExampleLedger
```

출력:

```
{
    "JournalS3Exports": [
        {
            "LedgerName": "myExampleLedger",
            "ExclusiveEndTime": 1568847599.0,
            "ExportCreationTime": 1568847801.418,
            "S3ExportConfiguration": {
                "Bucket": "awsExampleBucket",
                "Prefix": "ledgerexport1/",
                "EncryptionConfiguration": {
                    "ObjectEncryptionType": "SSE_S3"
                }
            },
            "ExportId": "ADR2ONPKN5LINYGb4dp7yZ",
            "RoleArn": "arn:aws:iam::123456789012:role/qldb-s3-export",
            "InclusiveStartTime": 1568764800.0,
            "Status": "IN_PROGRESS"
        }
```

```
    ]
}
```

자세한 내용은 Amazon 개발자 안내서의 Amazon에서 저널 내보내기QLDB를 참조하세요. QLDB

- API 자세한 내용은 AWS CLI 명령 참조의 ListJournalS3ExportsForLedger를 참조하세요.

### `list-journal-s3-exports`

다음 코드 예시에서는 `list-journal-s3-exports`을 사용하는 방법을 보여 줍니다.

AWS CLI

저널 내보내기 작업을 나열하려면

다음 `list-journal-s3-exports` 예제에서는 현재 AWS 계정 및 리전과 연결된 모든 원장에 대한 저널 내보내기 작업을 나열합니다.

```
aws qldb list-journal-s3-exports
```

출력:

```
{
    "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": "ADR2ONPKN5LINYGb4dp7yZ",
            "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
        }
    ]
}
```

자세한 내용은 Amazon 개발자 안내서의 Amazon에서 저널 내보내기QLDB를 참조하세요. QLDB

- API 자세한 내용은 AWS CLI 명령 참조의 ListJournalS3Exports를 참조하세요.

### `list-ledgers`

다음 코드 예시에서는 `list-ledgers`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 원장을 나열하는 방법

다음 `list-ledgers` 예제에서는 현재 AWS 계정 및 리전과 연결된 모든 원장을 나열합니다.

```
aws qldb list-ledgers
```

출력:

```
{
    "Ledgers": [
        {
            "State": "ACTIVE",
            "CreationDateTime": 1568839243.951,
            "Name": "myExampleLedger"
        },
        {
            "State": "ACTIVE",
```

```
            "CreationDateTime": 1568839543.557,
            "Name": "myExampleLedger2"
        }
    ]
}
```

자세한 내용은 Amazon 개발자 안내서의 Amazon QLDB Ledgers 기본 작업을 참조하세요. QLDB

- 자세한 API 내용은 명령 참조ListLedgers의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

원장에 연결된 태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 지정된 원장에 연결된 모든 태그를 나열합니다.

```
aws qldb list-tags-for-resource \
    --resource-arn arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger
```

출력:

```
{
    "Tags": {
        "IsTest": "true",
        "Domain": "Test"
    }
}
```

자세한 내용은 Amazon 개발자 안내서의 Amazon QLDB 리소스 태그 지정을 참조하세요. QLDB

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### `stream-journal-to-kinesis`

다음 코드 예시에서는 `stream-journal-to-kinesis`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 입력 파일을 사용하여 저널 데이터를 Kinesis Data Streams로 스트리밍하는 방법

다음 stream-journal-to-kinesis 예제에서는 이름이 인 원장에서 지정된 날짜 및 시간 범위 내에 저널 데이터 스트림을 생성합니다myExampleLedger. 스트림은 지정된 Amazon Kinesis 데이터 스트림으로 데이터를 전송합니다.

```
aws qldb stream-journal-to-kinesis \
    --ledger-name myExampleLedger \
    --inclusive-start-time 2020-05-29T00:00:00Z \
    --exclusive-end-time 2020-05-29T23:59:59Z \
    --role-arn arn:aws:iam::123456789012:role/my-kinesis-stream-role \
    --kinesis-configuration file://my-kinesis-config.json \
    --stream-name myExampleLedger-stream
```

my-kinesis-config.json의 콘텐츠:

```
{
    "StreamArn": "arn:aws:kinesis:us-east-1:123456789012:stream/stream-for-qldb",
    "AggregationEnabled": true
}
```

출력:

```
{
    "StreamId": "7ISCkqwe4y25YyHLzYUFAf"
}
```

자세한 내용은 Amazon 개발자 안내서의 Amazon에서 저널 데이터 스트리밍QLDB을 참조하세요. QLDB

예제 2: 단기 구문을 사용하여 저널 데이터를 Kinesis Data Streams로 스트리밍하려면

다음 stream-journal-to-kinesis 예제에서는 이름이 인 원장에서 지정된 날짜 및 시간 범위 내에 저널 데이터 스트림을 생성합니다myExampleLedger. 스트림은 지정된 Amazon Kinesis 데이터 스트림으로 데이터를 전송합니다.

```
aws qldb stream-journal-to-kinesis \
    --ledger-name myExampleLedger \
    --inclusive-start-time 2020-05-29T00:00:00Z \
    --exclusive-end-time 2020-05-29T23:59:59Z \
    --role-arn arn:aws:iam::123456789012:role/my-kinesis-stream-role \
    --stream-name myExampleLedger-stream \
```

```
    --kinesis-configuration StreamArn=arn:aws:kinesis:us-east-1:123456789012:stream/
stream-for-qldb,AggregationEnabled=true
```

출력:

```
{
    "StreamId": "7ISCkqwe4y25YyHLzYUFAf"
}
```

자세한 내용은 Amazon 개발자 안내서의 Amazon에서 저널 데이터 스트리밍QLDB을 참조하세요. QLDB

- 자세한 API 내용은 명령 참조StreamJournalToKinesis의 섹션을 참조하세요. AWS CLI

### tag-resource

다음 코드 예시에서는 tag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

원장에 태그를 지정하려면

다음 tag-resource 예제에서는 지정된 원장에 태그 세트를 추가합니다.

```
aws qldb tag-resource \
    --resource-arn arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger \
    --tags IsTest=true,Domain=Test
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon 개발자 안내서의 Amazon QLDB 리소스 태그 지정을 참조하세요. QLDB

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### untag-resource

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에서 태그를 제거하려면

다음 `untag-resource` 예제에서는 지정된 원장에서 지정된 태그 키가 있는 태그를 제거합니다.

```
aws qldb untag-resource \
    --resource-arn arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger \
    --tag-keys IsTest Domain
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon 개발자 안내서의 Amazon QLDB 리소스 태그 지정을 참조하세요. QLDB

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-ledger-permissions-mode

다음 코드 예시에서는 `update-ledger-permissions-mode`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 원장의 권한 모드를 로 업데이트하는 방법 STANDARD

다음 `update-ledger-permissions-mode` 예제에서는 지정된 원장에 `STANDARD` 권한 모드를 할당합니다.

```
aws qldb update-ledger-permissions-mode \
    --name myExampleLedger \
    --permissions-mode STANDARD
```

출력:

```
{
    "Name": "myExampleLedger",
    "Arn": "arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger",
    "PermissionsMode": "STANDARD"
}
```

예제 2: 원장의 권한 모드를 ALLOW_로 업데이트하는 방법ALL

다음 `update-ledger-permissions-mode` 예제에서는 지정된 원장에 `ALLOW_ALL` 권한 모드를 할당합니다.

```
aws qldb update-ledger-permissions-mode \
```

```
    --name myExampleLedger \
    --permissions-mode ALLOW_ALL
```

출력:

```
{
    "Name": "myExampleLedger",
    "Arn": "arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger",
    "PermissionsMode": "ALLOW_ALL"
}
```

자세한 내용은 Amazon 개발자 안내서의 Amazon QLDB Ledgers 기본 작업을 참조하세요. QLDB

- 자세한 API 내용은 명령 참조UpdateLedgerPermissionsMode의 섹션을 참조하세요. AWS CLI

### update-ledger

다음 코드 예시에서는 update-ledger을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 원장의 삭제 방지 속성을 업데이트하려면

다음 update-ledger 예제에서는 지정된 원장을 업데이트하여 삭제 보호 기능을 비활성화합니다.

```
aws qldb update-ledger \
    --name myExampleLedger \
    --no-deletion-protection
```

출력:

```
{
    "CreationDateTime": 1568839243.951,
    "Arn": "arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger",
    "DeletionProtection": false,
    "Name": "myExampleLedger",
    "State": "ACTIVE"
}
```

예제 2: 원장의 키를 고객 관리형 키로 업데이트 AWS KMS하는 방법

다음 update-ledger 예제에서는 지정된 원장을 업데이트하여 고객 관리형 KMS 키를 저장 시 암호화에 사용하도록 합니다.

```
aws qldb update-ledger \
    --name myExampleLedger \
    --kms-key arn:aws:kms:us-west-2:123456789012:key/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111
```

출력:

```
{
    "CreationDateTime": 1568839243.951,
    "Arn": "arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger",
    "DeletionProtection": false,
    "Name": "myExampleLedger",
    "State": "ACTIVE",
    "EncryptionDescription": {
        "KmsKeyArn": "arn:aws:kms:us-west-2:123456789012:key/a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111",
        "EncryptionStatus": "UPDATING"
    }
}
```

예제 3: 원장의 키를 AWS 소유 키로 업데이트 AWS KMS하는 방법

다음 update-ledger 예제에서는 지정된 원장을 업데이트하여 저장 시 암호화에 AWS 소유 KMS 키를 사용합니다.

```
aws qldb update-ledger \
    --name myExampleLedger \
    --kms-key AWS_OWNED_KMS_KEY
```

출력:

```
{
    "CreationDateTime": 1568839243.951,
    "Arn": "arn:aws:qldb:us-west-2:123456789012:ledger/myExampleLedger",
    "DeletionProtection": false,
    "Name": "myExampleLedger",
    "State": "ACTIVE",
    "EncryptionDescription": {
```

```
        "KmsKeyArn": "AWS_OWNED_KMS_KEY",
        "EncryptionStatus": "UPDATING"
    }
}
```

자세한 내용은 Amazon 개발자 안내서의 Amazon QLDB Ledgers 기본 작업을 참조하세요. QLDB

- 자세한 API 내용은 명령 참조UpdateLedger의 섹션을 참조하세요. AWS CLI

## 를 사용한 Amazon RDS 예제 AWS CLI

다음 코드 예제에서는 Amazon 에서 를 사용하여 작업을 수행하고 일반적인 시나리오 AWS Command Line Interface 를 구현하는 방법을 보여줍니다RDS.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `add-option-to-option-group`

다음 코드 예시에서는 `add-option-to-option-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

옵션 그룹에 옵션을 추가하려면

다음 `add-option-to-option-group` 예제에서는 지정된 옵션 그룹에 옵션을 추가합니다.

```
aws rds add-option-to-option-group \
    --option-group-name myoptiongroup \
    --options OptionName=OEM,Port=5500,DBSecurityGroupMemberships=default \
    --apply-immediately
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon RDS 사용 설명서의 옵션 그룹에 옵션 추가를 참조하세요.

- 자세한 API 내용은 명령 참조AddOptionToOptionGroup의 섹션을 참조하세요. AWS CLI

### add-role-to-db-cluster

다음 코드 예시에서는 add-role-to-db-cluster을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 자격 증명 및 액세스 관리(IAM) 역할을 DB 클러스터에 연결하려면

다음 add-role-to-db-cluster 예제에서는 역할을 DB 클러스터와 연결합니다.

```
aws rds add-role-to-db-cluster \
    --db-cluster-identifier mydbcluster \
    --role-arn arn:aws:iam::123456789012:role/RDSLoadFromS3
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL DB 클러스터와 IAM 역할 연결을 참조하세요.

- 자세한 API 내용은 명령 참조AddRoleToDbCluster의 섹션을 참조하세요. AWS CLI

### add-role-to-db-instance

다음 코드 예시에서는 add-role-to-db-instance을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS Identity and Access Management(IAM) 역할을 DB 인스턴스와 연결하려면

다음 add-role-to-db-instance 예제에서는 라는 Oracle DB 인스턴스에 역할을 추가합니다test-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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon 사용 설명서의 Amazon RDS Oracle과 Amazon S3 통합을 위한 사전 요구 사항을 참조하세요. RDS

- 자세한 API 내용은 명령 참조AddRoleToDbInstance의 섹션을 참조하세요. AWS CLI

### add-source-identifier-to-subscription

다음 코드 예시에서는 add-source-identifier-to-subscription을 사용하는 방법을 보여 줍니다.

AWS CLI

구독에 소스 식별자를 추가하려면

다음 add-source-identifier 예제에서는 기존 구독에 다른 소스 식별자를 추가합니다.

```
aws rds add-source-identifier-to-subscription \
    --subscription-name my-instance-events \
    --source-identifier test-instance-repl
```

출력:

```
{
    "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"
        ]
    }
}
```

- 자세한 API 내용은 명령 참조AddSourceIdentifierToSubscription의 섹션을 참조하세요. AWS CLI

### add-tags-to-resource

다음 코드 예시에서는 add-tags-to-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 태그를 추가하려면

다음 add-tags-to-resource 예제에서는 RDS 데이터베이스에 태그를 추가합니다.

```
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\"}]"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon 사용 설명서의 Amazon RDS 리소스 태그 지정을 참조하세요. RDS

- 자세한 API 내용은 명령 참조AddTagsToResource의 섹션을 참조하세요. AWS CLI

### apply-pending-maintenance-action

다음 코드 예시에서는 apply-pending-maintenance-action을 사용하는 방법을 보여 줍니다.

AWS CLI

보류되었던 유지 관리 작업을 적용하려면

다음 apply-pending-maintenance-action 예제에서는 DB 클러스터에 대해 보류 중인 유지 관리 작업을 적용합니다.

```
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
```

출력:

```
{
    "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"
            }
        ]
```

```
    }
}
```

자세한 내용은 Amazon RDS 사용 설명서의 DB 인스턴스 유지 관리 및 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 유지 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ApplyPendingMaintenanceAction의 섹션을 참조하세요. AWS CLI

### `authorize-db-security-group-ingress`

다음 코드 예시에서는 `authorize-db-security-group-ingress`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS Identity and Access Management(IAM) 역할을 DB 인스턴스와 연결하려면

다음 `authorize-db-security-group-ingress` 예제에서는 CIDR IP 범위 192.0.2.0/24에 대한 수신 규칙을 사용하여 기본 보안 그룹을 구성합니다.

```
aws rds authorize-db-security-group-ingress \
    --db-security-group-name default \
    --cidrip 192.0.2.0/24
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon RDS 사용 설명서의 IP 범위에서 DB 보안 그룹에 대한 네트워크 액세스 권한을 참조하세요.

- 자세한 API 내용은 명령 참조AuthorizeDbSecurityGroupIngress의 섹션을 참조하세요. AWS CLI

### backtrack-db-cluster

다음 코드 예시에서는 backtrack-db-cluster을 사용하는 방법을 보여 줍니다.

AWS CLI

Aurora DB 클러스터를 역추적하려면

다음 backtrack-db-cluster 예제는 지정된 DB 클러스터 샘플 클러스터를 2018년 3월 19일 오전 10시까지 역추적합니다.

```
aws rds backtrack-db-cluster --db-cluster-identifier sample-cluster --backtrack-
to 2018-03-19T10:00:00+00:00
```

이 명령은 RDS 리소스에 대한 변경을 확인하는 JSON 블록을 출력합니다.

- 자세한 API 내용은 명령 참조BacktrackDbCluster의 섹션을 참조하세요. AWS CLI

### cancel-export-task

다음 코드 예시에서는 cancel-export-task을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon S3로 스냅샷 내보내기를 취소하려면

다음 cancel-export-task 예제에서는 스냅샷을 Amazon S3로 내보내는 진행 중인 내보내기 작업을 취소합니다.

```
aws rds cancel-export-task \
    --export-task-identifier my-s3-export-1
```

출력:

```
{
    "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
}
```

자세한 내용은 Amazon 사용 설명서의 스냅샷 내보내기 작업 취소 또는 Amazon Aurora 사용 설명서의 스냅샷 내보내기 작업 취소를 참조하세요. RDS

- 자세한 API 내용은 명령 참조CancelExportTask의 섹션을 참조하세요. AWS CLI

### copy-db-cluster-parameter-group

다음 코드 예시에서는 copy-db-cluster-parameter-group을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 클러스터 파라미터 그룹을 복사하려면

다음 copy-db-cluster-parameter-group 예제에서는 DB 클러스터 파라미터 그룹의 복사본을 만듭니다.

```
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"
```

출력:

```
{
    "DBClusterParameterGroup": {
        "DBClusterParameterGroupName": "mydbclusterpgcopy",
        "DBClusterParameterGroupArn": "arn:aws:rds:us-east-1:123456789012:cluster-
pg:mydbclusterpgcopy",
```

```
        "DBParameterGroupFamily": "aurora-mysql5.7",
        "Description": "Copy of mydbclusterpg parameter group"
    }
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 DB 클러스터 파라미터 그룹 복사를 참조하세요.

- 자세한 API 내용은 명령 참조CopyDbClusterParameterGroup의 섹션을 참조하세요. AWS CLI

### copy-db-cluster-snapshot

다음 코드 예시에서는 copy-db-cluster-snapshot을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 클러스터 스냅샷을 복사하려면

다음 copy-db-cluster-snapshot 예제에서는 태그를 포함하여 DB 클러스터 스냅샷의 사본을 생성합니다.

```
aws rds copy-db-cluster-snapshot \
    --source-db-cluster-snapshot-identifier arn:aws:rds:us-
east-1:123456789012:cluster-snapshot:rds:myaurora-2019-06-04-09-16
    --target-db-cluster-snapshot-identifier myclustersnapshotcopy \
    --copy-tags
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 스냅샷 복사를 참조하세요.

- 자세한 API 내용은 명령 참조CopyDbClusterSnapshot의 섹션을 참조하세요. AWS CLI

### copy-db-parameter-group

다음 코드 예시에서는 copy-db-parameter-group을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 클러스터 파라미터 그룹을 복사하려면

다음 copy-db-parameter-group 예제에서는 DB 파라미터 그룹의 복사본을 만듭니다.

```
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"
```

출력:

```
{
    "DBParameterGroup": {
        "DBParameterGroupName": "mydbpgcopy",
        "DBParameterGroupArn": "arn:aws:rds:us-east-1:814387698303:pg:mydbpgcopy",
        "DBParameterGroupFamily": "mysql5.7",
        "Description": "Copy of mydbpg parameter group"
```

```
    }
}
```

자세한 내용은 Amazon RDS 사용 설명서의 DB 파라미터 그룹 복사를 참조하세요.

- 자세한 API 내용은 명령 참조CopyDbParameterGroup의 섹션을 참조하세요. AWS CLI

### copy-db-snapshot

다음 코드 예시에서는 copy-db-snapshot을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 스냅샷을 복사하려면

다음 copy-db-snapshot 예제에서는 DB 스냅샷의 사본을 생성합니다.

```
aws rds copy-db-snapshot \
    --source-db-snapshot-identifier rds:database-mysql-2019-06-06-08-38
    --target-db-snapshot-identifier mydbsnapshotcopy
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon RDS 사용 설명서의 스냅샷 복사를 참조하세요.

- 자세한 API 내용은 명령 참조CopyDbSnapshot의 섹션을 참조하세요. AWS CLI

### copy-option-group

다음 코드 예시에서는 copy-option-group을 사용하는 방법을 보여 줍니다.

AWS CLI

옵션 그룹을 복사하려면

다음 copy-option-group 예제에서는 옵션 그룹의 복사본을 만듭니다.

```
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"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon RDS 사용 설명서의 옵션 그룹 복사를 참조하세요.

- 자세한 API 내용은 명령 참조CopyOptionGroup의 섹션을 참조하세요. AWS CLI

### `create-blue-green-deployment`

다음 코드 예시에서는 `create-blue-green-deployment`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: for MySQL DB 인스턴스에 RDS 대한 블루/그린 배포를 생성하려면

다음 `create-blue-green-deployment` 예제에서는 MySQL DB 인스턴스에 대한 블루/그린 배포를 생성합니다.

```
aws rds create-blue-green-deployment \
    --blue-green-deployment-name bgd-cli-test-instance \
    --source arn:aws:rds:us-east-1:123456789012:db:my-db-instance \
    --target-engine-version 8.0 \
    --target-db-parameter-group-name mysql-80-group
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon RDS 사용 설명서의 블루/그린 배포 생성을 참조하세요.

예제 2: Aurora MySQL DB 클러스터에 대한 블루/그린 배포 생성

다음 `create-blue-green-deployment` 예제에서는 Aurora MySQL DB 클러스터에 대한 블루/그린 배포를 생성합니다.

```
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
```

출력:

```
{
      "BlueGreenDeployment": {
        "BlueGreenDeploymentIdentifier": "bgd-wi89nwzglccsfake",
        "BlueGreenDeploymentName": "my-blue-green-deployment",
        "Source": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-
cluster",
        "SwitchoverDetails": [
          {
            "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-
mysql-cluster",
            "Status": "PROVISIONING"
          },
          {
            "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-mysql-
cluster-1",
            "Status": "PROVISIONING"
          },
          {
            "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-mysql-
cluster-2",
            "Status": "PROVISIONING"
          },
          {
            "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-mysql-
cluster-3",
            "Status": "PROVISIONING"
          },
          {
            "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-endpoint:my-
excluded-member-endpoint",
            "Status": "PROVISIONING"
          },
          {
            "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-endpoint:my-
reader-endpoint",
            "Status": "PROVISIONING"
          }
        ],
        "Tasks": [
```

```
            {
              "Name": "CREATING_READ_REPLICA_OF_SOURCE",
              "Status": "PENDING"
            },
            {
              "Name": "DB_ENGINE_VERSION_UPGRADE",
              "Status": "PENDING"
            },
            {
              "Name": "CREATE_DB_INSTANCES_FOR_CLUSTER",
              "Status": "PENDING"
            },
            {
              "Name": "CREATE_CUSTOM_ENDPOINTS",
              "Status": "PENDING"
            }
          ],
          "Status": "PROVISIONING",
          "CreateTime": "2022-02-25T21:12:00.288000+00:00"
        }
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 블루/그린 배포 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateBlueGreenDeployment의 섹션을 참조하세요. AWS CLI

### `create-db-cluster-endpoint`

다음 코드 예시에서는 `create-db-cluster-endpoint`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 DB 클러스터 엔드포인트를 생성하려면

다음 `create-db-cluster-endpoint` 예제에서는 사용자 지정 DB 클러스터 엔드포인트를 생성하고 지정된 Aurora DB 클러스터와 연결합니다.

```
aws rds create-db-cluster-endpoint \
    --db-cluster-endpoint-identifier mycustomendpoint \
    --endpoint-type reader \
    --db-cluster-identifier mydbcluster \
    --static-members dbinstance1 dbinstance2
```

출력:

```
{
    "DBClusterEndpointIdentifier": "mycustomendpoint",
    "DBClusterIdentifier": "mydbcluster",
    "DBClusterEndpointResourceIdentifier": "cluster-endpoint-ANPAJ4AE5446DAEXAMPLE",
    "Endpoint": "mycustomendpoint.cluster-custom-cnpexample.us-
east-1.rds.amazonaws.com",
    "Status": "creating",
    "EndpointType": "CUSTOM",
    "CustomEndpointType": "READER",
    "StaticMembers": [
        "dbinstance1",
        "dbinstance2"
    ],
    "ExcludedMembers": [],
    "DBClusterEndpointArn": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:mycustomendpoint"
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora 연결 관리를 참조하세요.

- 자세한 API 내용은 명령 참조CreateDbClusterEndpoint의 섹션을 참조하세요. AWS CLI

### create-db-cluster-parameter-group

다음 코드 예시에서는 create-db-cluster-parameter-group을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 클러스터 파라미터 그룹을 생성하려면

다음 create-db-cluster-parameter-group 예제에서는 DB 클러스터 파라미터 그룹을 생성합니다.

```
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"
```

출력:

```
{
```

```
    "DBClusterParameterGroup": {
        "DBClusterParameterGroupName": "mydbclusterparametergroup",
        "DBParameterGroupFamily": "aurora5.6",
        "Description": "My new cluster parameter group",
        "DBClusterParameterGroupArn": "arn:aws:rds:us-east-1:123456789012:cluster-
pg:mydbclusterparametergroup"
    }
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 DB 클러스터 파라미터 그룹 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateDbClusterParameterGroup의 섹션을 참조하세요. AWS CLI

### `create-db-cluster-snapshot`

다음 코드 예시에서는 `create-db-cluster-snapshot`을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 클러스터 스냅샷을 생성하려면

다음 `create-db-cluster-snapshot` 예제에서는 DB 클러스터 스냅샷을 생성합니다.

```
aws rds create-db-cluster-snapshot \
    --db-cluster-identifier mydbcluster \
    --db-cluster-snapshot-identifier mydbclustersnapshot
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 DB 클러스터 스냅샷 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateDbClusterSnapshot의 섹션을 참조하세요. AWS CLI

### create-db-cluster

다음 코드 예시에서는 create-db-cluster을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: MySQL 5.7 호환 DB 클러스터 생성

다음 create-db-cluster 예제에서는 기본 엔진 버전을 사용하여 MySQL 5.7 호환 DB 클러스터를 생성합니다. 샘플 암호를 보안 암호secret99로 바꿉니다. 콘솔을 사용하여 DB 클러스터를 생성하면 Amazon은 DB 클러스터에 대한 라이터 DB 인스턴스를 RDS 자동으로 생성합니다. 그러나를 AWS CLI 사용하여 DB 클러스터를 생성할 때는 create-db-instance AWS CLI 명령을 사용하여 DB 클러스터에 대한 라이터 DB 인스턴스를 명시적으로 생성해야 합니다.

```
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
```

출력:


```
{
    "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-cnpexample.us-east-1.rds.amazonaws.com",
        "DBClusterParameterGroup": "default.aurora-mysql5.7",
        "HostedZoneId": "Z2R2ITUGPM61AM",
        "ReaderEndpoint": "sample-cluster.cluster-ro-cnpexample.us-
east-1.rds.amazonaws.com",
        "CopyTagsToSnapshot": false
```

```
    }
}
```

예제 2: Postgre SQL호환 DB 클러스터 생성

다음 `create-db-cluster` 예제에서는 기본 엔진 버전을 사용하여 Postgre SQL호환 DB 클러스터를 생성합니다. 예제 암호를 보안 암호`secret99`로 바꿉니다. 콘솔을 사용하여 DB 클러스터를 생성하면 Amazon은 DB 클러스터에 대한 라이터 DB 인스턴스를 RDS 자동으로 생성합니다. 그러나 를 AWS CLI 사용하여 DB 클러스터를 생성할 때는 `create-db-instance` AWS CLI 명령을 사용하여 DB 클러스터에 대한 라이터 DB 인스턴스를 명시적으로 생성해야 합니다.

```
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
```

출력:

```
{
    "DBCluster": {
        "Endpoint": "sample-pg-cluster.cluster-cnpexample.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-cnpexample.us-
east-1.rds.amazonaws.com",
        "MasterUsername": "master",
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1b",
            "us-east-1c"
        ],
        "ReadReplicaIdentifiers": [],
        "DBSubnetGroup": "default"
    }
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateDbCluster의 섹션을 참조하세요. AWS CLI

### `create-db-instance-read-replica`

다음 코드 예시에서는 `create-db-instance-read-replica`을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 인스턴스 읽기 전용 복제본을 생성하려면

이 예제에서는 라는 기존 DB 인스턴스의 읽기 전용 복제본을 생성합니다`test-instance`. 읽기 전용 복제본의 이름은 입니다`test-instance-repl`.

```
aws rds create-db-instance-read-replica \
    --db-instance-identifier test-instance-repl \
```

```
    --source-db-instance-identifier test-instance
```

출력:

```
{
    "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...
    }
}
```

- 자세한 API 내용은 명령 참조CreateDbInstanceReadReplica의 섹션을 참조하세요. AWS CLI

### create-db-instance

다음 코드 예시에서는 create-db-instance을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 인스턴스를 생성하려면

다음 create-db-instance 예제에서는 필수 옵션을 사용하여 새 DB 인스턴스를 시작합니다.

```
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
```

출력:

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "test-mysql-instance",
        "DBInstanceClass": "db.t3.micro",
        "Engine": "mysql",
```

```
        "DBInstanceStatus": "creating",
        "MasterUsername": "admin",
        "AllocatedStorage": 20,
        "PreferredBackupWindow": "12:55-13:25",
        "BackupRetentionPeriod": 1,
        "DBSecurityGroups": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-12345abc",
                "Status": "active"
            }
        ],
        "DBParameterGroups": [
            {
                "DBParameterGroupName": "default.mysql5.7",
                "ParameterApplyStatus": "in-sync"
            }
        ],
        "DBSubnetGroup": {
            "DBSubnetGroupName": "default",
            "DBSubnetGroupDescription": "default",
            "VpcId": "vpc-2ff2ff2f",
            "SubnetGroupStatus": "Complete",
            "Subnets": [
                {
                    "SubnetIdentifier": "subnet-########",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2c"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-########",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2d"
                    },
                    "SubnetStatus": "Active"
                },
                {
                    "SubnetIdentifier": "subnet-########",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2a"
                    },
                    "SubnetStatus": "Active"
```

```
                },
                {
                    "SubnetIdentifier": "subnet-########",
                    "SubnetAvailabilityZone": {
                        "Name": "us-west-2b"
                    },
                    "SubnetStatus": "Active"
                }
            ]
        },
        "PreferredMaintenanceWindow": "sun:08:07-sun:08:37",
        "PendingModifiedValues": {
            "MasterUserPassword": "****"
        },
        "MultiAZ": false,
        "EngineVersion": "5.7.22",
        "AutoMinorVersionUpgrade": true,
        "ReadReplicaDBInstanceIdentifiers": [],
        "LicenseModel": "general-public-license",
        "OptionGroupMemberships": [
            {
                "OptionGroupName": "default:mysql-5-7",
                "Status": "in-sync"
            }
        ],
        "PubliclyAccessible": true,
        "StorageType": "gp2",
        "DbInstancePort": 0,
        "StorageEncrypted": false,
        "DbiResourceId": "db-5555EXAMPLE44444444EXAMPLE",
        "CACertificateIdentifier": "rds-ca-2019",
        "DomainMemberships": [],
        "CopyTagsToSnapshot": false,
        "MonitoringInterval": 0,
        "DBInstanceArn": "arn:aws:rds:us-west-2:123456789012:db:test-mysql-
instance",
        "IAMDatabaseAuthenticationEnabled": false,
        "PerformanceInsightsEnabled": false,
        "DeletionProtection": false,
        "AssociatedRoles": []
    }
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon RDS DB 인스턴스 생성을 참조하세요. RDS

- 자세한 API 내용은 AWS CLI 명령 참조의 CreateDBInstance를 참조하세요.

### create-db-parameter-group

다음 코드 예시에서는 create-db-parameter-group을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 파라미터 그룹을 생성하려면

다음 create-db-parameter-group 예제에서는 DB 파라미터 그룹을 생성합니다.

```
aws rds create-db-parameter-group \
    --db-parameter-group-name mydbparametergroup \
    --db-parameter-group-family MySQL5.6 \
    --description "My new parameter group"
```

출력:

```
{
    "DBParameterGroup": {
        "DBParameterGroupName": "mydbparametergroup",
        "DBParameterGroupFamily": "mysql5.6",
        "Description": "My new parameter group",
        "DBParameterGroupArn": "arn:aws:rds:us-
east-1:123456789012:pg:mydbparametergroup"
    }
}
```

자세한 내용은 Amazon RDS 사용 설명서의 DB 파라미터 그룹 생성을 참조하세요.

- API 자세한 내용은 명령 참조의 C reateDBParameter그룹을 참조하세요. AWS CLI

### create-db-proxy-endpoint

다음 코드 예시에서는 create-db-proxy-endpoint을 사용하는 방법을 보여 줍니다.

AWS CLI

RDS 데이터베이스에 대한 DB 프록시 엔드포인트를 생성하려면

다음 create-db-proxy-endpoint 예제에서는 DB 프록시 엔드포인트를 생성합니다.

```
aws rds create-db-proxy-endpoint \
    --db-proxy-name proxyExample \
    --db-proxy-endpoint-name "proxyep1" \
    --vpc-subnet-ids subnetgroup1 subnetgroup2
```

출력:

```
{
"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",
        "CreatedDate": "2023-04-05T16:09:33.452000+00:00",
        "TargetRole": "READ_WRITE",
        "IsDefault": false
    }
}
```

자세한 내용은 Amazon RDS 사용 설명서의 프록시 엔드포인트 생성 및 Amazon Aurora 사용 설명서의 프록시 엔드포인트 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateDbProxyEndpoint의 섹션을 참조하세요. AWS CLI

### create-db-proxy

다음 코드 예시에서는 create-db-proxy을 사용하는 방법을 보여 줍니다.

AWS CLI

RDS 데이터베이스에 대한 DB 프록시를 생성하려면

다음 `create-db-proxy` 예제에서는 DB 프록시를 생성합니다.

```
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
```

출력:

```
{
"DBProxy": {
        "DBProxyName": "proxyExample",
        "DBProxyArn": "arn:aws:rds:us-east-1:123456789012:db-
proxy:prx-0123a01b12345c0ab",
        "EngineFamily": "MYSQL",
        "VpcId": "vpc-1234567",
        "VpcSecuritytGroupIds": [
            "sg-1234",
            "sg-5678",
            "sg-9101"
        ],
        "VpcSubnetIds": [
            "subnetgroup1",
            "subnetgroup2"
        ],
        "Auth": "[
            {
                "Description": "proxydescription1",
                "AuthScheme": "SECRETS",
                "SecretArn": "arn:aws:secretsmanager:us-
west-2:123456789123:secret:proxysecret1-Abcd1e",
                "IAMAuth": "DISABLED"
            }
        ]",
```

```
        "RoleArn": "arn:aws:iam::12345678912:role/ProxyRole",
        "Endpoint": "proxyExample.proxy-ab0cd1efghij.us-east-1.rds.amazonaws.com",
        "RequireTLS": false,
        "IdleClientTimeout": 1800,
        "DebuggingLogging": false,
        "CreatedDate": "2023-04-05T16:09:33.452000+00:00",
        "UpdatedDate": "2023-04-13T01:49:38.568000+00:00"
    }
}
```

자세한 내용은 Amazon RDS 사용 설명서의 RDS 프록시 생성 및 Amazon Aurora 사용 설명서의 RDS 프록시 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateDbProxy의 섹션을 참조하세요. AWS CLI

### `create-db-security-group`

다음 코드 예시에서는 `create-db-security-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon RDS DB 보안 그룹을 생성하려면

다음 `create-db-security-group` 명령은 새 Amazon RDS DB 보안 그룹을 생성합니다.

```
aws rds create-db-security-group --db-security-group-name mysecgroup --db-security-
group-description "My Test Security Group"
```

이 예제에서는 새 DB 보안 그룹의 이름이 지정mysecgroup되고 설명이 있습니다.

출력:

```
{
    "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"
    }
}
```

- 자세한 API 내용은 명령 참조CreateDbSecurityGroup의 섹션을 참조하세요. AWS CLI

### `create-db-shard-group`

다음 코드 예시에서는 `create-db-shard-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: Aurora PostgreSQL 기본 DB 클러스터 생성

다음 `create-db-cluster` 예제에서는 Aurora Serverless v2 및 Aurora Limitless Database와 호환되는 Aurora PostgreSQL SQL 기본 DB 클러스터를 생성합니다.

```
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
```

출력:

```
{
    "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-cekycexample.us-
east-2.rds.amazonaws.com",
        "ReaderEndpoint": "my-sv2-cluster.cluster-ro-cekycexample.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
```

```
        }
    }
}
```

예제 2: 기본(라이터) DB 인스턴스 생성

다음 `create-db-instance` 예제에서는 Aurora Serverless v2 기본(라이터) DB 인스턴스를 생성합니다. 콘솔을 사용하여 DB 클러스터를 생성하면 Amazon은 DB 클러스터에 대한 라이터 DB 인스턴스를 RDS 자동으로 생성합니다. 그러나 를 AWS CLI 사용하여 DB 클러스터를 생성할 때는 `create-db-instance` AWS CLI 명령을 사용하여 DB 클러스터에 대한 라이터 DB 인스턴스를 명시적으로 생성해야 합니다.

```
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
```

출력:

```
{
    "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
    }
}
```

예제 3: DB 샤드 그룹 생성

다음 `create-db-shard-group` 예제에서는 Aurora PostgreSQL 기본 DB 클러스터에 DB 샤드 그룹을 생성합니다.

```
aws rds create-db-shard-group \
    --db-shard-group-identifier my-db-shard-group \
```

```
    --db-cluster-identifier my-sv2-cluster \
    --max-acu 768
```

출력:

```
{
    "DBShardGroupResourceId": "shardgroup-a6e3a0226aa243e2ac6c7a1234567890",
    "DBShardGroupIdentifier": "my-db-shard-group",
    "DBClusterIdentifier": "my-sv2-cluster",
    "MaxACU": 768.0,
    "ComputeRedundancy": 0,
    "Status": "creating",
    "PubliclyAccessible": false,
    "Endpoint": "my-sv2-cluster.limitless-cekycexample.us-east-2.rds.amazonaws.com"
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 Aurora Serverless v2 사용을 참조하세요.

- 자세한 API 내용은 명령 참조CreateDbShardGroup의 섹션을 참조하세요. AWS CLI

### `create-db-snapshot`

다음 코드 예시에서는 `create-db-snapshot`을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 스냅샷을 생성하려면

다음 예제에서는 DB 스냅샷을 생성합니다.

```
aws rds create-db-snapshot \
    --db-instance-identifier database-mysql \
    --db-snapshot-identifier mydbsnapshot
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon RDS 사용 설명서의 DB 스냅샷 생성을 참조하세요.

- 자세한 API 내용은 AWS CLI 명령 참조의 CreateDBSnapshot를 참조하세요.

### create-db-subnet-group

다음 코드 예시에서는 create-db-subnet-group을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 서브넷 그룹을 생성하려면

다음 create-db-subnet-group 예제에서는 기존 서브넷mysubnetgroup을 사용하여 라는 DB 서브넷 그룹을 생성합니다.

```
aws rds create-db-subnet-group \
    --db-subnet-group-name mysubnetgroup \
    --db-subnet-group-description "test DB subnet group" \
    --subnet-ids
 '["subnet-0a1dc4e1a6f123456","subnet-070dd7ecb3aaaaaaa","subnet-00f5b198bc0abcdef"]'
```

출력:

```
{
    "DBSubnetGroup": {
        "DBSubnetGroupName": "mysubnetgroup",
        "DBSubnetGroupDescription": "test DB subnet group",
        "VpcId": "vpc-0f08e7610a1b2c3d4",
        "SubnetGroupStatus": "Complete",
        "Subnets": [
            {
                "SubnetIdentifier": "subnet-070dd7ecb3aaaaaaa",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2b"
                },
                "SubnetStatus": "Active"
            },
            {
                "SubnetIdentifier": "subnet-00f5b198bc0abcdef",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2d"
                },
                "SubnetStatus": "Active"
            },
            {
                "SubnetIdentifier": "subnet-0a1dc4e1a6f123456",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2b"
                },
                "SubnetStatus": "Active"
            }
        ],
        "DBSubnetGroupArn": "arn:aws:rds:us-
west-2:0123456789012:subgrp:mysubnetgroup"
    }
}
```

자세한 내용은 Amazon RDS 사용 설명서의 에서 DB 인스턴스 생성을 VPC 참조하세요.

- 자세한 API 내용은 명령 참조CreateDbSubnetGroup의 섹션을 참조하세요. AWS CLI

### `create-event-subscription`

다음 코드 예시에서는 `create-event-subscription`을 사용하는 방법을 보여 줍니다.

AWS CLI

이벤트 구독을 생성하려면

다음 `create-event-subscription` 예제에서는 현재 AWS 계정의 DB 인스턴스에 대한 백업 및 복구 이벤트에 대한 구독을 생성합니다. 알림은 에서 지정한 Amazon Simple Notification Service 주제로 전송됩니다`--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
```

출력:

```
{
    "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
    }
}
```

- 자세한 API 내용은 명령 참조CreateEventSubscription의 섹션을 참조하세요. AWS CLI

## `create-global-cluster`

다음 코드 예시에서는 `create-global-cluster`을 사용하는 방법을 보여 줍니다.

AWS CLI

글로벌 DB 클러스터를 생성하려면

다음 `create-global-cluster` 예제에서는 새 Aurora My SQL호환 글로벌 DB 클러스터를 생성합니다.

```
aws rds create-global-cluster \
    --global-cluster-identifier myglobalcluster \
    --engine aurora-mysql
```

출력:

```
{
    "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": []
    }
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 Aurora 글로벌 데이터베이스 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateGlobalCluster의 섹션을 참조하세요. AWS CLI

### `create-option-group`

다음 코드 예시에서는 `create-option-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon RDS 옵션 그룹을 생성하려면

다음 `create-option-group` 명령은 `Oracle Enterprise Edition` 버전에 대한 새 Amazon RDS 옵션 그룹을 생성하고 설명을 `11.2`, is named ``MyOptionGroup` 포함합니다.

```
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"
```

출력:

```
{
    "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"
    }
}
```

- 자세한 API 내용은 명령 참조CreateOptionGroup의 섹션을 참조하세요. AWS CLI

### delete-blue-green-deployment

다음 코드 예시에서는 delete-blue-green-deployment을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: RDS for MySQL DB 인스턴스에 대한 녹색 환경의 리소스를 삭제하려면

다음 delete-blue-green-deployment 예제에서는 RDS for MySQL DB 인스턴스에 대한 그린 환경의 리소스를 삭제합니다.

```
aws rds delete-blue-green-deployment \
    --blue-green-deployment-identifier bgd-v53303651eexfake \
    --delete-target
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon RDS 사용 설명서의 블루/그린 배포 삭제를 참조하세요.

예제 2: Aurora MySQL DB 클러스터의 녹색 환경에서 리소스를 삭제하려면

다음 `delete-blue-green-deployment` 예제에서는 Aurora MySQL DB 클러스터에 대한 그린 환경의 리소스를 삭제합니다.

```
aws rds delete-blue-green-deployment \
    --blue-green-deployment-identifier bgd-wi89nwzglccsfake \
    --delete-target
```

출력:

```
{
    "BlueGreenDeployment": {
        "BlueGreenDeploymentIdentifier": "bgd-wi89nwzglccsfake",
        "BlueGreenDeploymentName": "my-blue-green-deployment",
        "Source": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-
cluster",
        "Target": "arn:aws:rds:us-east-1:123456789012:cluster:my-aurora-mysql-
cluster-green-3rnukl",
        "SwitchoverDetails": [
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-
aurora-mysql-cluster",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster:my-
aurora-mysql-cluster-green-3rnukl",
                "Status": "AVAILABLE"
```

```
            },
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-1",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-1-green-gpmaxf",
                "Status": "AVAILABLE"
            },
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-2",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-2-green-j2oajq",
                "Status": "AVAILABLE"
            },
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-3",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:db:my-aurora-
mysql-cluster-3-green-mkxies",
                "Status": "AVAILABLE"
            },
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-excluded-member-endpoint",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-excluded-member-endpoint-green-4sqjrq",
                "Status": "AVAILABLE"
            },
            {
                "SourceMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-reader-endpoint",
                "TargetMember": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:my-reader-endpoint-green-gwwzlg",
                "Status": "AVAILABLE"
            }
        ],
        "Tasks": [
            {
                "Name": "CREATING_READ_REPLICA_OF_SOURCE",
                "Status": "COMPLETED"
            },
            {
                "Name": "DB_ENGINE_VERSION_UPGRADE",
```

```
                "Status": "COMPLETED"
            },
            {
                "Name": "CREATE_DB_INSTANCES_FOR_CLUSTER",
                "Status": "COMPLETED"
            },
            {
                "Name": "CREATE_CUSTOM_ENDPOINTS",
                "Status": "COMPLETED"
            }
        ],
        "Status": "DELETING",
        "CreateTime": "2022-02-25T21:12:00.288000+00:00",
        "DeleteTime": "2022-02-25T22:29:11.336000+00:00"
    }
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 블루/그린 배포 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteBlueGreenDeployment의 섹션을 참조하세요. AWS CLI

### `delete-db-cluster-endpoint`

다음 코드 예시에서는 `delete-db-cluster-endpoint`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 DB 클러스터 엔드포인트를 삭제하려면

다음 `delete-db-cluster-endpoint` 예제에서는 지정된 사용자 지정 DB 클러스터 엔드포인트를 삭제합니다.

```
aws rds delete-db-cluster-endpoint \
    --db-cluster-endpoint-identifier mycustomendpoint
```

출력:

```
{
    "DBClusterEndpointIdentifier": "mycustomendpoint",
    "DBClusterIdentifier": "mydbcluster",
    "DBClusterEndpointResourceIdentifier": "cluster-endpoint-ANPAJ4AE5446DAEXAMPLE",
    "Endpoint": "mycustomendpoint.cluster-custom-cnpexample.us-
east-1.rds.amazonaws.com",
```

```
    "Status": "deleting",
    "EndpointType": "CUSTOM",
    "CustomEndpointType": "READER",
    "StaticMembers": [
        "dbinstance1",
        "dbinstance2",
        "dbinstance3"
    ],
    "ExcludedMembers": [],
    "DBClusterEndpointArn": "arn:aws:rds:us-east-1:123456789012:cluster-
endpoint:mycustomendpoint"
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora 연결 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteDbClusterEndpoint의 섹션을 참조하세요. AWS CLI

### `delete-db-cluster-parameter-group`

다음 코드 예시에서는 `delete-db-cluster-parameter-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 클러스터 파라미터 그룹을 삭제하려면

다음 `delete-db-cluster-parameter-group` 예제에서는 지정된 DB 클러스터 파라미터 그룹을 삭제합니다.

```
aws rds delete-db-cluster-parameter-group \
    --db-cluster-parameter-group-name mydbclusterparametergroup
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Aurora 사용 설명서의 DB 파라미터 그룹 및 DB 클러스터 파라미터 그룹 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteDbClusterParameterGroup의 섹션을 참조하세요. AWS CLI

### `delete-db-cluster-snapshot`

다음 코드 예시에서는 `delete-db-cluster-snapshot`을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 클러스터 스냅샷을 삭제하려면

다음 delete-db-cluster-snapshot 예제에서는 지정된 DB 클러스터 스냅샷을 삭제합니다.

```
aws rds delete-db-cluster-snapshot \
    --db-cluster-snapshot-identifier mydbclustersnapshot
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 스냅샷 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteDbClusterSnapshot의 섹션을 참조하세요. AWS CLI

### delete-db-cluster

다음 코드 예시에서는 delete-db-cluster을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: DB 클러스터에서 DB 인스턴스를 삭제하려면

다음 delete-db-instance 예제에서는 DB 클러스터의 최종 DB 인스턴스를 삭제합니다. 삭제 상태가 아닌 DB 인스턴스가 포함된 DB 클러스터는 삭제할 수 없습니다. DB 클러스터에서 DB 인스턴스를 삭제할 때는 최종 스냅샷을 만들 수 없습니다.

```
aws rds delete-db-instance \
    --db-instance-identifier database-3
```

출력:

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "database-3",
        "DBInstanceClass": "db.r4.large",
        "Engine": "aurora-postgresql",
        "DBInstanceStatus": "deleting",

    ...output omitted...

    }
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 Aurora DB 클러스터에서 DB 인스턴스 삭제를 참조하세요.

예제 2: DB 클러스터 삭제

다음 delete-db-cluster 예제에서는 라는 DB 클러스터를 삭제mycluster하고 라는 최종 스냅샷을 생성합니다mycluster-final-snapshot. 스냅샷을 가져오는 동안 DB 클러스터의 상태를 사용할 수 있습니다. 삭제 진행 상황을 추적하려면 describe-db-clusters CLI 명령을 사용합니다.

```
aws rds delete-db-cluster \
    --db-cluster-identifier mycluster \
```

```
    --no-skip-final-snapshot \
    --final-db-snapshot-identifier mycluster-final-snapshot
```

출력:

```
{
    "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...

    }
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 단일 DB 인스턴스가 있는 Aurora 클러스터를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteDbCluster의 섹션을 참조하세요. AWS CLI

### delete-db-instance-automated-backup

다음 코드 예시에서는 delete-db-instance-automated-backup을 사용하는 방법을 보여 줍니다.

AWS CLI

리전에서 복제된 자동 백업을 삭제하려면

다음 delete-db-instance-automated-backup 예제에서는 지정된 Amazon 리소스 이름()을 사용하여 자동 백업을 삭제합니다ARN.

```
aws rds delete-db-instance-automated-backup \
```

```
    --db-instance-automated-backups-arn "arn:aws:rds:us-west-2:123456789012:auto-
backup:ab-jkib2gfq5rv7replzadausbrktni2bn4example"
```

출력:

```
{
    "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-jkib2gfq5rv7replzadausbrktni2bn4example"
    }
}
```

자세한 내용은 Amazon RDS 사용 설명서의 복제된 백업 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteDbInstanceAutomatedBackup의 섹션을 참조하세요. AWS CLI

### `delete-db-instance`

다음 코드 예시에서는 `delete-db-instance`을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 인스턴스를 삭제하려면

다음 delete-db-instance 예제에서는 test-instance-final-snap이라는 최종 DB 스냅샷을 만든 후 지정된 DB 인스턴스를 삭제합니다.

```
aws rds delete-db-instance \
    --db-instance-identifier test-instance \
    --final-db-snapshot-identifier test-instance-final-snap
```

출력:

```
{
    "DBInstance": {
        "DBInstanceIdentifier": "test-instance",
        "DBInstanceStatus": "deleting",
        ...some output truncated...
    }
}
```

- 자세한 API 내용은 AWS CLI 명령 참조의 DeleteDBInstance를 참조하세요.

### delete-db-parameter-group

다음 코드 예시에서는 delete-db-parameter-group을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 파라미터 그룹을 삭제하려면

다음 command 예제에서는 DB 파라미터 그룹을 삭제합니다.

```
aws rds delete-db-parameter-group \
    --db-parameter-group-name mydbparametergroup
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon RDS 사용 설명서의 DB 파라미터 그룹 작업을 참조하세요.

- API 자세한 내용은 명령 참조의 D eleteDBParameter그룹을 참조하세요. AWS CLI

### delete-db-proxy-endpoint

다음 코드 예시에서는 delete-db-proxy-endpoint을 사용하는 방법을 보여 줍니다.

AWS CLI

RDS 데이터베이스에 대한 DB 프록시 엔드포인트를 삭제하려면

다음 delete-db-proxy-endpoint 예제에서는 대상 데이터베이스에 대한 DB 프록시 엔드포인트를 삭제합니다.

```
aws rds delete-db-proxy-endpoint \
    --db-proxy-endpoint-name proxyEP1
```

출력:

```
{
"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",
        "CreatedDate": "2023-04-13T01:49:38.568000+00:00",
        "TargetRole": "READ_ONLY",
        "IsDefault": false
    }
}
```

자세한 내용은 Amazon RDS 사용 설명서의 프록시 엔드포인트 삭제 및 Amazon Aurora 사용 설명서의 프록시 엔드포인트 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteDbProxyEndpoint의 섹션을 참조하세요. AWS CLI

### delete-db-proxy

다음 코드 예시에서는 delete-db-proxy을 사용하는 방법을 보여 줍니다.

AWS CLI

RDS 데이터베이스의 DB 프록시를 삭제하려면

다음 delete-db-proxy 예제에서는 DB 프록시를 삭제합니다.

```
aws rds delete-db-proxy \
    --db-proxy-name proxyExample
```

출력:

```
{
        "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,
        "CreatedDate": "2023-04-05T16:09:33.452000+00:00",
        "UpdatedDate": "2023-04-13T01:49:38.568000+00:00"
    }
}
```

자세한 내용은 Amazon RDS 사용 설명서의 RDS 프록시 삭제 및 Amazon Aurora 사용 설명서의 RDS 프록시 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteDbProxy의 섹션을 참조하세요. AWS CLI

### delete-db-security-group

다음 코드 예시에서는 delete-db-security-group을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 보안 그룹을 삭제하려면

다음 delete-db-security-group 예제에서는 라는 DB 보안 그룹을 삭제합니다mysecuritygroup.

```
aws rds delete-db-security-group \
    --db-security-group-name mysecuritygroup
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon RDS 사용 설명서의 DB 보안 그룹 작업(EC2-Classic 플랫폼)을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteDbSecurityGroup의 섹션을 참조하세요. AWS CLI

### delete-db-shard-group

다음 코드 예시에서는 delete-db-shard-group을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: DB 샤드 그룹을 삭제하지 못했습니다.

다음 delete-db-shard-group 예제에서는 모든 데이터베이스와 스키마를 삭제하기 전에 DB 샤드 그룹을 삭제하려고 할 때 발생하는 오류를 보여줍니다.

```
aws rds delete-db-shard-group \
    --db-shard-group-identifier limitless-test-shard-grp
```

출력:

```
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.
```

예제 2: DB 샤드 그룹을 성공적으로 삭제하려면

다음 delete-db-shard-group 예제에서는 스키마를 포함한 모든 데이터베이스 및 스키마를 삭제한 후 DB public 샤드 그룹을 삭제합니다.

```
aws rds delete-db-shard-group \
    --db-shard-group-identifier limitless-test-shard-grp
```

출력:

```
{
    "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-cekycexample.us-
east-2.rds.amazonaws.com"
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 Aurora DB 클러스터 및 DB 인스턴스 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteDbShardGroup의 섹션을 참조하세요. AWS CLI

### delete-db-snapshot

다음 코드 예시에서는 delete-db-snapshot을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 스냅샷을 삭제하려면

다음 delete-db-snapshot 예제에서는 지정된 DB 스냅샷을 삭제합니다.

```
aws rds delete-db-snapshot \
    --db-snapshot-identifier mydbsnapshot
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon RDS 사용 설명서의 스냅샷 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteDbSnapshot의 섹션을 참조하세요. AWS CLI

### delete-db-subnet-group

다음 코드 예시에서는 delete-db-subnet-group을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 서브넷 그룹을 삭제하려면

다음 delete-db-subnet-group 예제에서는 라는 DB 서브넷 그룹을 삭제합니다mysubnetgroup.

```
aws rds delete-db-subnet-group --db-subnet-group-name mysubnetgroup
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon RDS 사용 설명서의 에서 DB 인스턴스 작업을 VPC 참조하세요.

- 자세한 API 내용은 명령 참조DeleteDbSubnetGroup의 섹션을 참조하세요. AWS CLI

### delete-event-subscription

다음 코드 예시에서는 delete-event-subscription을 사용하는 방법을 보여 줍니다.

AWS CLI

이벤트 구독을 삭제하려면

다음 delete-event-subscription 예제에서는 지정된 이벤트 구독을 삭제합니다.

```
aws rds delete-event-subscription --subscription-name my-instance-events
```

출력:

```
{
    "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"
    }
}
```

- 자세한 API 내용은 명령 참조DeleteEventSubscription의 섹션을 참조하세요. AWS CLI

### `delete-global-cluster`

다음 코드 예시에서는 `delete-global-cluster`을 사용하는 방법을 보여 줍니다.

AWS CLI

글로벌 DB 클러스터를 삭제하려면

다음 `delete-global-cluster` 예제에서는 Aurora My SQL호환 글로벌 DB 클러스터를 삭제합니다. 출력에는 삭제하려는 클러스터가 표시되지만 후속 `describe-global-clusters` 명령에는 해당 DB 클러스터가 나열되지 않습니다.

```
aws rds delete-global-cluster \
    --global-cluster-identifier myglobalcluster
```

출력:

```
{
    "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": []
    }
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 Aurora 글로벌 데이터베이스 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteGlobalCluster의 섹션을 참조하세요. AWS CLI

### delete-option-group

다음 코드 예시에서는 delete-option-group을 사용하는 방법을 보여 줍니다.

AWS CLI

옵션 그룹을 삭제하려면

다음 delete-option-group 예제에서는 지정된 옵션 그룹을 삭제합니다.

```
aws rds delete-option-group \
    --option-group-name myoptiongroup
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon RDS 사용 설명서의 옵션 그룹 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteOptionGroup의 섹션을 참조하세요. AWS CLI

### deregister-db-proxy-targets

다음 코드 예시에서는 deregister-db-proxy-targets을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터베이스 대상 그룹에서 DB 프록시 대상을 등록 취소하려면

다음 deregister-db-proxy-targets 예제에서는 프록시proxyExample와 대상 간의 연결을 제거합니다.

```
aws rds deregister-db-proxy-targets \
    --db-proxy-name proxyExample \
```

```
    --db-instance-identifiers database-1
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon RDS 사용 설명서의 RDS 프록시 삭제 및 Amazon Aurora 사용 설명서의 RDS 프록시 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeregisterDbProxyTargets의 섹션을 참조하세요. AWS CLI

### describe-account-attributes

다음 코드 예시에서는 `describe-account-attributes`을 사용하는 방법을 보여 줍니다.

AWS CLI

계정 속성을 설명하려면

다음 `describe-account-attributes` 예제에서는 현재 AWS 계정의 속성을 검색합니다.

```
aws rds describe-account-attributes
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeAccountAttributes의 섹션을 참조하세요. AWS CLI

### describe-blue-green-deployments

다음 코드 예시에서는 describe-blue-green-deployments을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 생성이 완료된 후 RDS DB 인스턴스의 블루/그린 배포 설명

다음 describe-blue-green-deployment 예제에서는 생성이 완료된 후 블루/그린 배포의 세부 정보를 검색합니다.

```
aws rds describe-blue-green-deployments \
    --blue-green-deployment-identifier bgd-v53303651eexfake
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon RDS 사용 설명서의 블루/그린 배포 보기를 참조하세요.

예제 2: Aurora MySQL DB 클러스터에 대한 블루/그린 배포 설명

다음 `describe-blue-green-deployment` 예제에서는 블루/그린 배포의 세부 정보를 검색합니다.

```
aws rds describe-blue-green-deployments \
    --blue-green-deployment-identifier bgd-wi89nwzglccsfake
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 블루/그린 배포 보기를 참조하세요.

예제 3: 전환 후 Aurora MySQL 클러스터에 대한 블루/그린 배포 설명

다음 describe-blue-green-deployment 예제에서는 그린 환경이 프로덕션 환경으로 승격된 후 블루/그린 배포에 대한 세부 정보를 검색합니다.

```
aws rds describe-blue-green-deployments \
    --blue-green-deployment-identifier bgd-wi89nwzglccsfake
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 블루/그린 배포 보기를 참조하세요.

예제 4: 결합된 블루/그린 배포 설명

다음 `describe-blue-green-deployment` 예제에서는 결합된 블루/그린 배포의 세부 정보를 검색합니다.

```
aws rds describe-blue-green-deployments
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon RDS 사용 설명서의 블루/그린 배포 보기 및 Amazon Aurora 사용 설명서의 블루/그린 배포 보기를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeBlueGreenDeployments의 섹션을 참조하세요. AWS CLI

### `describe-certificates`

다음 코드 예시에서는 `describe-certificates`을 사용하는 방법을 보여 줍니다.

AWS CLI

인증서를 설명하려면

다음 describe-certificates 예제에서는 사용자의 기본 리전과 연결된 인증서의 세부 정보를 검색합니다.

```
aws rds describe-certificates
```

출력:

```
{
    "Certificates": [
        {
            "CertificateIdentifier": "rds-ca-ecc384-g1",
            "CertificateType": "CA",
            "Thumbprint": "2ee3dcc06e50192559b13929e73484354f23387d",
            "ValidFrom": "2021-05-24T22:06:59+00:00",
            "ValidTill": "2121-05-24T23:06:59+00:00",
            "CertificateArn": "arn:aws:rds:us-west-2::cert:rds-ca-ecc384-g1",
            "CustomerOverride": false
        },
        {
            "CertificateIdentifier": "rds-ca-rsa4096-g1",
            "CertificateType": "CA",
            "Thumbprint": "19da4f2af579a8ae1f6a0fa77aa5befd874b4cab",
            "ValidFrom": "2021-05-24T22:03:20+00:00",
            "ValidTill": "2121-05-24T23:03:20+00:00",
            "CertificateArn": "arn:aws:rds:us-west-2::cert:rds-ca-rsa4096-g1",
            "CustomerOverride": false
        },
        {
            "CertificateIdentifier": "rds-ca-rsa2048-g1",
            "CertificateType": "CA",
            "Thumbprint": "7c40cb42714b6fdb2b296f9bbd0e8bb364436a76",
            "ValidFrom": "2021-05-24T21:59:00+00:00",
            "ValidTill": "2061-05-24T22:59:00+00:00",
            "CertificateArn": "arn:aws:rds:us-west-2::cert:rds-ca-rsa2048-g1",
            "CustomerOverride": true,
            "CustomerOverrideValidTill": "2061-05-24T22:59:00+00:00"
        },
        {
            "CertificateIdentifier": "rds-ca-2019",
            "CertificateType": "CA",
            "Thumbprint": "d40ddb29e3750dffa671c3140bbf5f478d1c8096",
            "ValidFrom": "2019-08-22T17:08:50+00:00",
            "ValidTill": "2024-08-22T17:08:50+00:00",
```

```
            "CertificateArn": "arn:aws:rds:us-west-2::cert:rds-ca-2019",
            "CustomerOverride": false
        }
    ],
    "DefaultCertificateForNewLaunches": "rds-ca-rsa2048-g1"
}
```

자세한 내용은 Amazon RDS 사용 설명서의 SSL/TLS를 사용하여 DB 인스턴스에 대한 연결 암호화 및 Amazon Aurora 사용 설명서의 SSL/TLS를 사용하여 DB 클러스터에 대한 연결을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeCertificates의 섹션을 참조하세요. AWS CLI

### describe-db-cluster-backtracks

다음 코드 예시에서는 describe-db-cluster-backtracks을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 클러스터의 역추적을 설명하려면

다음 describe-db-cluster-backtracks 예제에서는 지정된 DB 클러스터에 대한 세부 정보를 검색합니다.

```
aws rds describe-db-cluster-backtracks \
    --db-cluster-identifier mydbcluster
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 Aurora DB 클러스터 역추적을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeDbClusterBacktracks의 섹션을 참조하세요. AWS CLI

### describe-db-cluster-endpoints

다음 코드 예시에서는 describe-db-cluster-endpoints을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: DB 클러스터 엔드포인트 설명

다음 describe-db-cluster-endpoints 예제에서는 DB 클러스터 엔드포인트에 대한 세부 정보를 검색합니다. 가장 일반적인 종류의 Aurora 클러스터에는 두 개의 엔드포인트가 있습니다. 하나의 엔드포인트에는 유형이 있습니다WRITER. 이 엔드포인트를 모든 SQL 문에 사용할 수 있습니다. 다른 엔드포인트에는 유형이 있습니다READER. 이 엔드포인트는 SELECT 및 기타 읽기 전용 SQL 문에만 사용할 수 있습니다.

```
aws rds describe-db-cluster-endpoints
```

출력:

```
{
    "DBClusterEndpoints": [
        {
            "DBClusterIdentifier": "my-database-1",
            "Endpoint": "my-database-1.cluster-cnpexample.us-
east-1.rds.amazonaws.com",
            "Status": "creating",
            "EndpointType": "WRITER"
        },
        {
            "DBClusterIdentifier": "my-database-1",
            "Endpoint": "my-database-1.cluster-ro-cnpexample.us-
east-1.rds.amazonaws.com",
            "Status": "creating",
```

```
            "EndpointType": "READER"
        },
        {
            "DBClusterIdentifier": "mydbcluster",
            "Endpoint": "mydbcluster.cluster-cnpexamle.us-east-1.rds.amazonaws.com",
            "Status": "available",
            "EndpointType": "WRITER"
        },
        {
            "DBClusterIdentifier": "mydbcluster",
            "Endpoint": "mydbcluster.cluster-ro-cnpexample.us-
east-1.rds.amazonaws.com",
            "Status": "available",
            "EndpointType": "READER"
        }
    ]
}
```

예제 2: 단일 DB 클러스터의 DB 클러스터 엔드포인트 설명

다음 `describe-db-cluster-endpoints` 예제에서는 지정된 단일 DB 클러스터의 DB 클러스터 엔드포인트에 대한 세부 정보를 검색합니다. Aurora Serverless 클러스터에는 유형이 인 단일 엔드포인트만 있습니다`WRITER`.

```
aws rds describe-db-cluster-endpoints \
    --db-cluster-identifier serverless-cluster
```

출력:

```
{
    "DBClusterEndpoints": [
        {
            "Status": "available",
            "Endpoint": "serverless-cluster.cluster-cnpexample.us-
east-1.rds.amazonaws.com",
            "DBClusterIdentifier": "serverless-cluster",
            "EndpointType": "WRITER"
        }
    ]
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora 연결 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeDbClusterEndpoints의 섹션을 참조하세요. AWS CLI

### describe-db-cluster-parameter-groups

다음 코드 예시에서는 describe-db-cluster-parameter-groups을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 클러스터 파라미터 그룹을 설명하려면

다음 describe-db-cluster-parameter-groups 예제에서는 DB 클러스터 파라미터 그룹에 대한 세부 정보를 검색합니다.

```
aws rds describe-db-cluster-parameter-groups
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 DB 파라미터 그룹 및 DB 클러스터 파라미터 그룹 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeDbClusterParameterGroups의 섹션을 참조하세요. AWS CLI

### `describe-db-cluster-parameters`

다음 코드 예시에서는 `describe-db-cluster-parameters`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: DB 클러스터 파라미터 그룹의 파라미터를 설명하려면

다음 `describe-db-cluster-parameters` 예제에서는 DB 클러스터 파라미터 그룹의 파라미터에 대한 세부 정보를 검색합니다.

```
aws rds describe-db-cluster-parameters \
    --db-cluster-parameter-group-name mydbclusterpg
```

출력:

```
{
    "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...
    ]
}
```

예제 2: DB 클러스터 파라미터 그룹의 파라미터 이름만 나열하려면

다음 `describe-db-cluster-parameters` 예제에서는 DB 클러스터 파라미터 그룹의 파라미터 이름만 검색합니다.

```
aws rds describe-db-cluster-parameters \
    --db-cluster-parameter-group-name default.aurora-mysql5.7 \
    --query 'Parameters[].{ParameterName:ParameterName}'
```

출력:

```
[
    {
```

```
        "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...
    }
]
```

예제 3: DB 클러스터 파라미터 그룹에서 수정 가능한 파라미터만 설명하려면

다음 `describe-db-cluster-parameters` 예제에서는 DB 클러스터 파라미터 그룹에서 수정할 수 있는 파라미터의 이름만 검색합니다.

```
aws rds describe-db-cluster-parameters \
    --db-cluster-parameter-group-name default.aurora-mysql5.7 \
    --query 'Parameters[].{ParameterName:ParameterName,IsModifiable:IsModifiable} |
 [?IsModifiable == `true`]'
```

출력:

```
[
    {
        "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...
    }
]
```

예제 4: DB 클러스터 파라미터 그룹에서 수정 가능한 부울 파라미터만 설명하려면

다음 describe-db-cluster-parameters 예제에서는 DB 클러스터 파라미터 그룹에서 수정할 수 있고 부울 데이터 유형이 있는 파라미터의 이름만 검색합니다.

```
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`]'
```

출력:

```
[
    {
        "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...
    }
]
```

자세한 내용은 Amazon Aurora 사용 설명서의 DB 파라미터 그룹 및 DB 클러스터 파라미터 그룹 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeDbClusterParameters의 섹션을 참조하세요. AWS CLI

### `describe-db-cluster-snapshot-attributes`

다음 코드 예시에서는 `describe-db-cluster-snapshot-attributes`을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 클러스터 스냅샷의 속성 이름 및 값을 설명하려면

다음 `describe-db-cluster-snapshot-attributes` 예제에서는 지정된 DB 클러스터 스냅샷의 속성 이름 및 값에 대한 세부 정보를 검색합니다.

```
aws rds describe-db-cluster-snapshot-attributes \
    --db-cluster-snapshot-identifier myclustersnapshot
```

출력:

```
{
    "DBClusterSnapshotAttributesResult": {
        "DBClusterSnapshotIdentifier": "myclustersnapshot",
        "DBClusterSnapshotAttributes": [
            {
                "AttributeName": "restore",
                "AttributeValues": [
                    "123456789012"
                ]
            }
        ]
    }
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 DB 클러스터 스냅샷 공유를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeDbClusterSnapshotAttributes의 섹션을 참조하세요. AWS CLI

### `describe-db-cluster-snapshots`

다음 코드 예시에서는 `describe-db-cluster-snapshots`을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 클러스터에 대한 DB 클러스터 스냅샷을 설명하려면

다음 `describe-db-cluster-snapshots` 예제에서는 지정된 DB 클러스터의 DB 클러스터 스냅샷에 대한 세부 정보를 검색합니다.

```
aws rds describe-db-cluster-snapshots \
    --db-cluster-identifier mydbcluster
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 DB 클러스터 스냅샷 생성을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeDbClusterSnapshots의 섹션을 참조하세요. AWS CLI

### `describe-db-clusters`

다음 코드 예시에서는 `describe-db-clusters`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: DB 클러스터 설명

다음 `describe-db-clusters` 예제에서는 지정된 DB 클러스터의 세부 정보를 검색합니다.

```
aws rds describe-db-clusters \
    --db-cluster-identifier mydbcluster
```

출력:

```
{
    "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-cnpexample.us-
east-1.rds.amazonaws.com",
            "ReaderEndpoint": "mydbcluster.cluster-ro-cnpexample.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:mydbcluster",
            "AssociatedRoles": [],
            "IAMDatabaseAuthenticationEnabled": false,
            "ClusterCreateTime": "2019-04-15T14:18:42.785Z",
            "EngineMode": "provisioned",
            "DeletionProtection": false,
            "HttpEndpointEnabled": false
        }
    ]
}
```

예제 2: 모든 DB 클러스터의 특정 속성을 나열하려면

다음 `describe-db-clusters` 예제에서는 현재 AWS 리전에 있는 모든 DB 클러스터의 `DBClusterIdentifier`, `Endpoint` 및 `ReaderEndpoint` 속성만 검색합니다.

```
aws rds describe-db-clusters \
    --query 'DBClusters[].
{DBClusterIdentifier:DBClusterIdentifier,Endpoint:Endpoint,ReaderEndpoint:ReaderEndpoint}'
```

출력:

```
[
    {
        "Endpoint": "cluster-57-2020-05-01-2270.cluster-cnpexample.us-
east-1.rds.amazonaws.com",
        "ReaderEndpoint": "cluster-57-2020-05-01-2270.cluster-ro-cnpexample.us-
east-1.rds.amazonaws.com",
        "DBClusterIdentifier": "cluster-57-2020-05-01-2270"
    },
    {
        "Endpoint": "cluster-57-2020-05-01-4615.cluster-cnpexample.us-
east-1.rds.amazonaws.com",
        "ReaderEndpoint": "cluster-57-2020-05-01-4615.cluster-ro-cnpexample.us-
east-1.rds.amazonaws.com",
        "DBClusterIdentifier": "cluster-57-2020-05-01-4615"
    },
    {
```

```
        "Endpoint": "pg2-cluster.cluster-cnpexample.us-east-1.rds.amazonaws.com",
        "ReaderEndpoint": "pg2-cluster.cluster-ro-cnpexample.us-
east-1.rds.amazonaws.com",
        "DBClusterIdentifier": "pg2-cluster"
    },
    ...output omitted...
    }
]
```

예제 3: 특정 속성이 있는 DB 클러스터를 나열하려면

다음 `describe-db-clusters` 예제에서는 DB 엔진을 사용하는 `aurora-postgresql` DB 클러스터의 `DBClusterIdentifier` 및 `Engine` 속성만 검색합니다.

```
aws rds describe-db-clusters \
    --query 'DBClusters[].{DBClusterIdentifier:DBClusterIdentifier,Engine:Engine} |
 [?Engine == `aurora-postgresql`]'
```

출력:

```
[
    {
        "Engine": "aurora-postgresql",
        "DBClusterIdentifier": "pg2-cluster"
    }
]
```

자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeDbClusters의 섹션을 참조하세요. AWS CLI

### `describe-db-engine-versions`

다음 코드 예시에서는 `describe-db-engine-versions`을 사용하는 방법을 보여 줍니다.

AWS CLI

MySQL DB 엔진의 DB 엔진 버전 설명

다음 `describe-db-engine-versions` 예제는 지정된 DB 엔진의 각 DB 엔진 버전에 대한 세부 정보를 표시합니다.

```
aws rds describe-db-engine-versions \
    --engine mysql
```

출력:

```
{
    "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...
            ]
        }
```

자세한 내용은 Amazon 사용 설명서의 Amazon Relational Database Service(AmazonRDS)란 무엇입니까?를 참조하세요. RDS

- API 자세한 내용은 AWS CLI 명령 참조의 D escribeDBEngine버전을 참조하세요.

### describe-db-instance-automated-backups

다음 코드 예시에서는 describe-db-instance-automated-backups을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 인스턴스의 자동 백업을 설명하려면

다음 describe-db-instance-automated-backups 예제에서는 지정된 DB 인스턴스의 자동 백업에 대한 세부 정보를 표시합니다. 세부 정보에는 다른 AWS 리전의 복제된 자동 백업이 포함됩니다.

```
aws rds describe-db-instance-automated-backups \
    --db-instance-identifier new-orcl-db
```

출력:

```
{
    "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-jkib2gfq5rv7replzadausbrktni2bn4example"
        }
    ]
}
```

자세한 내용은 Amazon RDS 사용 설명서의 복제된 백업에 대한 정보 찾기를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeDbInstanceAutomatedBackups의 섹션을 참조하세요. AWS CLI

### describe-db-instances

다음 코드 예시에서는 describe-db-instances을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 인스턴스를 설명하려면

다음 describe-db-instances 예제에서는 지정된 DB 인스턴스에 대한 세부 정보를 검색합니다.

```
aws rds describe-db-instances \
    --db-instance-identifier mydbinstancecf
```

출력:

```
{
    "DBInstances": [
        {
            "DBInstanceIdentifier": "mydbinstancecf",
            "DBInstanceClass": "db.t3.small",
            "Engine": "mysql",
            "DBInstanceStatus": "available",
            "MasterUsername": "masterawsuser",
            "Endpoint": {
                "Address": "mydbinstancecf.abcexample.us-east-1.rds.amazonaws.com",
                "Port": 3306,
                "HostedZoneId": "Z2R2ITUGPM61AM"
            },
            ...some output truncated...
        }
    ]
```

```
}
```

- 자세한 API 내용은 AWS CLI 명령 참조의 DescribeDBInstances를 참조하세요.

### `describe-db-log-files`

다음 코드 예시에서는 `describe-db-log-files`을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 인스턴스의 로그 파일을 설명하려면

다음 `describe-db-log-files` 예제에서는 지정된 DB 인스턴스의 로그 파일에 대한 세부 정보를 검색합니다.

```
aws rds describe-db-log-files -\
    -db-instance-identifier test-instance
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeDbLogFiles의 섹션을 참조하세요. AWS CLI

### `describe-db-parameter-groups`

다음 코드 예시에서는 `describe-db-parameter-groups`을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 파라미터 그룹을 설명하려면

다음 `describe-db-parameter-groups` 예제에서는 DB 파라미터 그룹에 대한 세부 정보를 검색합니다.

```
aws rds describe-db-parameter-groups
```

출력:

```
{
    "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...
    ]
}
```

자세한 내용은 Amazon RDS 사용 설명서의 DB 파라미터 그룹 작업을 참조하세요.

- API 자세한 내용은 AWS CLI 명령 참조의 D escribeDBParameter그룹을 참조하세요.

### `describe-db-parameters`

다음 코드 예시에서는 `describe-db-parameters`을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 파라미터 그룹의 파라미터를 설명하려면

다음 `describe-db-parameters` 예제에서는 지정된 DB 파라미터 그룹의 세부 정보를 검색합니다.

```
aws rds describe-db-parameters \
    --db-parameter-group-name mydbpg
```

출력:

```
{
```

```
    "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...
    ]
}
```

자세한 내용은 Amazon RDS 사용 설명서의 DB 파라미터 그룹 작업을 참조하세요.

- 자세한 API 내용은 AWS CLI 명령 참조의 DescribeDBParameters를 참조하세요.

### `describe-db-proxies`

다음 코드 예시에서는 `describe-db-proxies`을 사용하는 방법을 보여 줍니다.

AWS CLI

RDS 데이터베이스에 대한 DB 프록시를 설명하려면

다음 `describe-db-proxies` 예제에서는 DB 프록시에 대한 정보를 반환합니다.

```
aws rds describe-db-proxies
```

출력:

```
{
    "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,
            "CreatedDate": "2023-04-05T16:09:33.452000+00:00",
            "UpdatedDate": "2023-04-13T01:49:38.568000+00:00"
        },
        {
            "DBProxyName": "proxyExample2",
            "DBProxyArn": "arn:aws:rds:us-east-1:123456789012:db-
proxy:prx-1234a12b23456c1ab",
            "Status": "available",
            "EngineFamily": "PostgreSQL",
            "VpcId": "sg-1234567",
            "VpcSecurityGroupIds": [
```

```
                "sg-1234"
            ],
            "VpcSubnetIds": [
                "subnetgroup1",
                "subnetgroup2"
            ],
            "Auth": "[
                {
                    "Description": "proxydescription2"
                    "AuthScheme": "SECRETS",
                    "SecretArn": "aarn:aws:secretsmanager:us-
west-2:123456789123:secret:secretName-1234f",
                    "IAMAuth": "DISABLED"
                }
            ]",
            "RoleArn": "arn:aws:iam::12345678912:role/ProxyPostgreSQLRole",
            "Endpoint": "proxyExample2.proxy-ab0cd1efghij.us-
east-1.rds.amazonaws.com",
            "RequireTLS": false,
            "IdleClientTimeout": 1800,
            "DebuggingLogging": false,
            "CreatedDate": "2022-01-05T16:19:33.452000+00:00",
            "UpdatedDate": "2023-04-13T01:49:38.568000+00:00"
        }
    ]
}
```

자세한 내용은 Amazon RDS 사용 설명서의 RDS 프록시 보기 및 Amazon Aurora 사용 설명서의 RDS 프록시 보기를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeDbProxies의 섹션을 참조하세요. AWS CLI

### `describe-db-proxy-endpoints`

다음 코드 예시에서는 `describe-db-proxy-endpoints`을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 프록시 엔드포인트를 설명하려면

다음 `describe-db-proxy-endpoints` 예제에서는 DB 프록시 엔드포인트에 대한 정보를 반환합니다.

```
aws rds describe-db-proxy-endpoints
```

출력:

```
{
    "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",
            "CreatedDate": "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",
            "CreatedDate": "2023-04-05T16:09:33.452000+00:00",
```

```
            "TargetRole": "READ_WRITE",
            "IsDefault": false
        }
    ]
}
```

자세한 내용은 Amazon RDS 사용 설명서의 프록시 엔드포인트 보기 및 Amazon Aurora 사용 설명서의 프록시 엔드포인트 생성을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeDbProxyEndpoints의 섹션을 참조하세요. AWS CLI

### `describe-db-proxy-target-groups`

다음 코드 예시에서는 `describe-db-proxy-target-groups`을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 프록시 엔드포인트를 설명하려면

다음 `describe-db-proxy-target-groups` 예제에서는 DB 프록시 대상 그룹에 대한 정보를 반환합니다.

```
aws rds describe-db-proxy-target-groups \
    --db-proxy-name proxyExample
```

출력:

```
{
"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": []
        },
        "CreatedDate": "2023-05-02T18:41:19.495000+00:00",
```

```
        "UpdatedDate": "2023-05-02T18:41:21.762000+00:00"
    }
}
```

자세한 내용은 Amazon RDS 사용 설명서의 RDS 프록시 보기 및 Amazon Aurora 사용 설명서의 RDS 프록시 보기를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeDbProxyTargetGroups의 섹션을 참조하세요. AWS CLI

### `describe-db-proxy-targets`

다음 코드 예시에서는 `describe-db-proxy-targets`을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 프록시 대상을 설명하려면

다음 `describe-db-proxy-targets` 예제에서는 DB 프록시 대상에 대한 정보를 반환합니다.

```
aws rds describe-db-proxy-targets \
    --db-proxy-name proxyExample
```

출력:

```
{
    "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"
            }
        }
    ]
}
```

자세한 내용은 Amazon RDS 사용 설명서의 RDS 프록시 보기 및 Amazon Aurora 사용 설명서의 RDS 프록시 보기를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeDbProxyTargets의 섹션을 참조하세요. AWS CLI

### `describe-db-recommendations`

다음 코드 예시에서는 `describe-db-recommendations`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 모든 DB 권장 사항 나열

다음 `describe-db-recommendations` 예제에서는 AWS 계정의 모든 DB 권장 사항을 나열합니다.

```
aws rds describe-db-recommendations
```

출력:

```
{
    "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"
                }
            ]
        }
    ]
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon RDS 권장 사항 보기 및 응답과 Amazon Aurora 사용 설명서의 Amazon RDS 권장 사항 보기 및 응답을 참조하세요. RDS

예제 2: 심각도가 높은 DB 권장 사항 나열

다음 `describe-db-recommendations` 예제에서는 AWS 계정의 심각도가 높은 DB 권장 사항을 나열합니다.

```
aws rds describe-db-recommendations \
    --filters Name=severity,Values=high
```

출력:

```
{
    "DBRecommendations": [
        {
            "RecommendationId": "12ab3cde-f456-7g8h-9012-i3j45678k9lm",
            "TypeId": "config_recommendation::rds_extended_support",
            "Severity": "high",
            "ResourceArn": "arn:aws:rds:us-west-2:111122223333:db:database-1",
            "Status": "active",
            "CreatedTime": "2024-02-21T23:14:19.392000+00:00",
            "UpdatedTime": "2024-02-21T23:14:19+00:00",
            "Detection": "Your databases will be auto-enrolled to RDS Extended
 Support on February 29",
            "Recommendation": "Upgrade your major version before February 29, 2024
 to avoid additional charges",
            "Description": "Your PostgreSQL 11 and MySQL 5.7 databases will be
 automatically enrolled into RDS Extended Support on February 29, 2024. To avoid
```

```
 the increase in charges due to RDS Extended Support, we recommend upgrading your
 databases to a newer major engine version before February 29, 2024.\nTo learn more
 about the RDS Extended Support pricing, refer to the pricing page.",
            "RecommendedActions": [
                {
                    "ActionId": "12ab34c5de6fg7h89i0jk1lm234n5678",
                    "Parameters": [],
                    "ApplyModes": [
                        "manual"
                    ],
                    "Status": "ready",
                    "ContextAttributes": []
                }
            ],
            "Category": "cost optimization",
            "Source": "RDS",
            "TypeDetection": "Your database will be auto-enrolled to RDS Extended
 Support on February 29",
            "TypeRecommendation": "Upgrade your major version before February 29,
 2024 to avoid additional charges",
            "Impact": "Increase in charges due to RDS Extended Support",
            "AdditionalInfo": "With Amazon RDS Extended Support, you can continue
 running your database on a major engine version past the RDS end of standard
 support date for an additional cost. This paid feature gives you more time to
 upgrade to a supported major engine version.\nDuring Extended Support, Amazon RDS
 will supply critical CVE patches and bug fixes.",
            "Links": [
                {
                    "Text": "Amazon RDS Extended Support pricing for RDS for MySQL",
                    "Url": "https://aws.amazon.com/rds/mysql/pricing/"
                },
                {
                    "Text": "Amazon RDS Extended Support for RDS for MySQL and
 PostgreSQL databases",
                    "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/
extended-support.html"
                },
                {
                    "Text": "Amazon RDS Extended Support pricing for Amazon Aurora
 PostgreSQL",
                    "Url": "https://aws.amazon.com/rds/aurora/pricing/"
                },
                {
```

```
                    "Text": "Amazon RDS Extended Support for Aurora PostgreSQL
 databases",
                    "Url": "https://docs.aws.amazon.com/AmazonRDS/latest/
AuroraUserGuide/extended-support.html"
                },
                {
                    "Text": "Amazon RDS Extended Support pricing for RDS for
 PostgreSQL",
                    "Url": "https://aws.amazon.com/rds/postgresql/pricing/"
                }
            ]
        }
    ]
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon RDS 권장 사항 보기 및 응답과 Amazon Aurora 사용 설명서의 Amazon RDS 권장 사항 보기 및 응답을 참조하세요. RDS

예제 3: 지정된 DB 인스턴스에 대한 DB 권장 사항을 나열하려면

다음 `describe-db-recommendations` 예제에서는 지정된 DB 인스턴스에 대한 모든 DB 권장 사항을 나열합니다.

```
aws rds describe-db-recommendations \
    --filters Name=dbi-resource-id,Values=database-1
```

출력:

```
{
    "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"
                }
            ]
        }
    ]
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon RDS 권장 사항 보기 및 응답과 Amazon Aurora 사용 설명서의 Amazon RDS 권장 사항 보기 및 응답을 참조하세요. RDS

예제 4: 모든 활성 DB 권장 사항을 나열하려면

다음 `describe-db-recommendations` 예제에서는 AWS 계정의 모든 활성 DB 권장 사항을 나열합니다.

```
aws rds describe-db-recommendations \
    --filters Name=status,Values=active
```

출력:

```
{
    "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"
                }
            ]
        }
    ]
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon RDS 권장 사항 보기 및 응답과 Amazon Aurora 사용 설명서의 Amazon RDS 권장 사항 보기 및 응답을 참조하세요. RDS

- 자세한 API 내용은 명령 참조DescribeDbRecommendations의 섹션을 참조하세요. AWS CLI

### `describe-db-security-groups`

다음 코드 예시에서는 `describe-db-security-groups`을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 보안 그룹을 나열하려면

다음 `describe-db-security-groups` 예제에서는 DB 보안 그룹을 나열합니다.

```
aws rds describe-db-security-groups
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon RDS 사용 설명서의 사용 가능한 DB 보안 그룹 나열을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeDbSecurityGroups의 섹션을 참조하세요. AWS CLI

### `describe-db-shard-groups`

다음 코드 예시에서는 `describe-db-shard-groups`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: DB 샤드 그룹 설명

다음 describe-db-shard-groups 예제에서는 DB 샤드 그룹의 세부 정보를 검색합니다.

```
aws rds describe-db-shard-groups
```

출력:

```
{
    "DBShardGroups": [
        {
            "DBShardGroupResourceId": "shardgroup-7bb446329da94788b3f957746example",
            "DBShardGroupIdentifier": "limitless-test-shard-grp",
            "DBClusterIdentifier": "limitless-test-cluster",
            "MaxACU": 768.0,
            "ComputeRedundancy": 0,
            "Status": "available",
            "PubliclyAccessible": true,
            "Endpoint": "limitless-test-cluster.limitless-cekycexample.us-
east-2.rds.amazonaws.com"
        },
        {
            "DBShardGroupResourceId": "shardgroup-a6e3a0226aa243e2ac6c7a1234567890",
            "DBShardGroupIdentifier": "my-db-shard-group",
            "DBClusterIdentifier": "my-sv2-cluster",
            "MaxACU": 768.0,
            "ComputeRedundancy": 0,
            "Status": "available",
            "PubliclyAccessible": false,
            "Endpoint": "my-sv2-cluster.limitless-cekycexample.us-
east-2.rds.amazonaws.com"
        }
    ]
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeDbShardGroups의 섹션을 참조하세요. AWS CLI

### describe-db-snapshot-attributes

다음 코드 예시에서는 describe-db-snapshot-attributes을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 스냅샷의 속성 이름 및 값을 설명하려면

다음 describe-db-snapshot-attributes 예제에서는 DB 스냅샷의 속성 이름과 값을 설명합니다.

```
aws rds describe-db-snapshot-attributes \
    --db-snapshot-identifier mydbsnapshot
```

출력:

```
{
    "DBSnapshotAttributesResult": {
        "DBSnapshotIdentifier": "mydbsnapshot",
        "DBSnapshotAttributes": [
            {
                "AttributeName": "restore",
                "AttributeValues": [
                    "123456789012",
                    "210987654321"
                ]
            }
        ]
    }
}
```

자세한 내용은 Amazon RDS 사용 설명서의 DB 스냅샷 공유를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeDbSnapshotAttributes의 섹션을 참조하세요. AWS CLI

### describe-db-snapshots

다음 코드 예시에서는 describe-db-snapshots을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: DB 인스턴스의 DB 스냅샷을 설명하려면

다음 describe-db-snapshots 예제에서는 DB 인스턴스의 DB 스냅샷 세부 정보를 검색합니다.

```
aws rds describe-db-snapshots \
    --db-snapshot-identifier mydbsnapshot
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon RDS 사용 설명서의 DB 스냅샷 생성을 참조하세요.

예 2: 생성한 수동 스냅샷의 개수를 확인하려면

다음 describe-db-snapshots 예제에서는 --query 옵션의 length 연산자를 사용하여 특정 AWS 리전에서 가져온 수동 스냅샷 수를 반환합니다.

```
aws rds describe-db-snapshots \
    --snapshot-type manual \
    --query "length(*[].{DBSnapshots:SnapshotType})" \
    --region eu-central-1
```

출력:

```
35
```

자세한 내용은 Amazon RDS 사용 설명서의 DB 스냅샷 생성을 참조하세요.

- 자세한 API 내용은 AWS CLI 명령 참조의 DescribeDBSnapshots를 참조하세요.

### describe-db-subnet-groups

다음 코드 예시에서는 describe-db-subnet-groups을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 서브넷 그룹을 설명하려면

다음 describe-db-subnet-groups 예제에서는 지정된 DB 서브넷 그룹의 세부 정보를 검색합니다.

```
aws rds describe-db-subnet-groups
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon Virtual Private Cloud VPCs 및 RDS Amazon을 참조하세요.  RDS

- 자세한 API 내용은 명령 참조DescribeDbSubnetGroups의 섹션을 참조하세요. AWS CLI

### `describe-engine-default-cluster-parameters`

다음 코드 예시에서는 `describe-engine-default-cluster-parameters`을 사용하는 방법을 보여 줍니다.

AWS CLI

Aurora 데이터베이스 엔진의 기본 엔진 및 시스템 파라미터 정보를 설명하려면

다음 describe-engine-default-cluster-parameters 예제에서는 MySQL 5.7과 호환되는 Aurora DB 클러스터에 대한 기본 엔진 및 시스템 파라미터 정보의 세부 정보를 검색합니다.

```
aws rds describe-engine-default-cluster-parameters \
    --db-parameter-group-family aurora-mysql5.7
```

출력:

```
{
    "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...
        ]
    }
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 DB 파라미터 그룹 및 DB 클러스터 파라미터 그룹 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeEngineDefaultClusterParameters의 섹션을 참조하세요. AWS CLI

### describe-engine-default-parameters

다음 코드 예시에서는 describe-engine-default-parameters을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터베이스 엔진의 기본 엔진 및 시스템 파라미터 정보를 설명하려면

다음 describe-engine-default-parameters 예제에서는 MySQL 5.7 DB 인스턴스의 기본 엔진 및 시스템 파라미터 정보에 대한 세부 정보를 검색합니다.

```
aws rds describe-engine-default-parameters \
    --db-parameter-group-family mysql5.7
```

출력:

```
{
    "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...
        ]
    }
}
```

자세한 내용은 Amazon RDS 사용 설명서의 DB 파라미터 그룹 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeEngineDefaultParameters의 섹션을 참조하세요. AWS CLI

### describe-event-categories

다음 코드 예시에서는 describe-event-categories을 사용하는 방법을 보여 줍니다.

AWS CLI

이벤트 범주를 설명하려면

다음 describe-event-categories 예제에서는 사용 가능한 모든 이벤트 소스의 이벤트 범주에 대한 세부 정보를 검색합니다.

```
aws rds describe-event-categories
```

출력:

```
{
    "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"
            ]
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeEventCategories의 섹션을 참조하세요. AWS CLI

### `describe-event-subscriptions`

다음 코드 예시에서는 `describe-event-subscriptions`을 사용하는 방법을 보여 줍니다.

AWS CLI

이벤트 구독을 설명하려면

이 예제에서는 현재 AWS 계정에 대한 모든 Amazon RDS 이벤트 구독을 설명합니다.

```
aws rds describe-event-subscriptions
```

출력:

```
{
    "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...
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeEventSubscriptions의 섹션을 참조하세요. AWS CLI

### `describe-events`

다음 코드 예시에서는 `describe-events`을 사용하는 방법을 보여 줍니다.

AWS CLI

이벤트를 설명하려면

다음 `describe-events` 예제에서는 지정된 DB 인스턴스에 대해 발생한 이벤트의 세부 정보를 검색합니다.

```
aws rds describe-events \
    --source-identifier test-instance \
    --source-type db-instance
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeEvents의 섹션을 참조하세요. AWS CLI

### describe-export-tasks

다음 코드 예시에서는 describe-export-tasks을 사용하는 방법을 보여 줍니다.

AWS CLI

스냅샷 내보내기 작업을 설명하려면

다음 describe-export-tasks 예제에서는 스냅샷 내보내기에 대한 정보를 Amazon S3로 반환합니다.

```
aws rds describe-export-tasks
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 Amazon RDS 사용 설명서의 스냅샷 내보내기 모니터링을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeExportTasks의 섹션을 참조하세요. AWS CLI

### describe-global-clusters

다음 코드 예시에서는 describe-global-clusters을 사용하는 방법을 보여 줍니다.

AWS CLI

글로벌 DB 클러스터를 설명하려면

다음 describe-global-clusters 예제에서는 현재 AWS 리전의 Aurora 글로벌 DB 클러스터를 나열합니다.

```
aws rds describe-global-clusters
```

출력:

```
{
    "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": []
        }
    ]
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 Aurora 글로벌 데이터베이스 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeGlobalClusters의 섹션을 참조하세요. AWS CLI

### describe-option-group-options

다음 코드 예시에서는 describe-option-group-options을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 모든 옵션을 설명하려면

다음 describe-option-group-options 예제에서는 Oracle Database 19c 인스턴스에 대한 두 가지 옵션을 나열합니다.

```
aws rds describe-option-group-options \
    --engine-name oracle-ee \
    --major-engine-version 19 \
    --max-items 2
```

출력:

```
{
    "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=="
```

```
}
```

자세한 내용은 Amazon RDS 사용 설명서의 옵션 그룹에 대한 옵션 및 옵션 설정 나열을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeOptionGroupOptions의 섹션을 참조하세요. AWS CLI

### `describe-option-groups`

다음 코드 예시에서는 `describe-option-groups`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 옵션 그룹을 설명하려면

다음 `describe-option-groups` 예제에서는 Oracle Database 19c 인스턴스의 옵션 그룹을 나열합니다.

```
aws rds describe-option-groups \
    --engine-name oracle-ee \
    --major-engine-version 19
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon RDS 사용 설명서의 옵션 그룹에 대한 옵션 및 옵션 설정 나열을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeOptionGroups의 섹션을 참조하세요. AWS CLI

### describe-orderable-db-instance-options

다음 코드 예시에서는 describe-orderable-db-instance-options을 사용하는 방법을 보여 줍니다.

AWS CLI

주문 가능한 DB 인스턴스 옵션을 설명하려면

다음 describe-orderable-db-instance-options 예제에서는 MySQL DB 엔진을 실행하는 DB 인스턴스의 주문 가능한 옵션에 대한 세부 정보를 검색합니다.

```
aws rds describe-orderable-db-instance-options \
    --engine mysql
```

출력:

```
{
    "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...
}
```

- 자세한 API 내용은 명령 참조DescribeOrderableDBInstanceOptions의 섹션을 참조하세요. AWS CLI

### describe-pending-maintenance-actions

다음 코드 예시에서는 describe-pending-maintenance-actions을 사용하는 방법을 보여 줍니다.

AWS CLI

하나 이상의 보류 중인 유지 관리 작업이 있는 리소스를 나열하려면

다음 describe-pending-maintenance-actions 예제에서는 DB 인스턴스에 대해 보류 중인 유지 관리 작업을 나열합니다.

```
aws rds describe-pending-maintenance-actions
```

출력:

```
{
    "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"
                }
            ]
```

```
        }
    ]
}
```

자세한 내용은 Amazon RDS 사용 설명서의 DB 인스턴스 유지 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DescribePendingMaintenanceActions의 섹션을 참조하세요. AWS CLI

### `describe-reserved-db-instances-offerings`

다음 코드 예시에서는 `describe-reserved-db-instances-offerings`을 사용하는 방법을 보여 줍니다.

AWS CLI

예약 DB 인스턴스 제공 설명

다음 `describe-reserved-db-instances-offerings` 예제에서는 에 대해 예약된 DB 인스턴스 옵션에 대한 세부 정보를 검색합니다`oracle`.

```
aws rds describe-reserved-db-instances-offerings \
    --product-description oracle
```

출력:

```
{
    "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...
}
```

- 자세한 API 내용은 명령 참조DescribeReservedDbInstancesOfferings의 섹션을 참조하세요. AWS CLI

### `describe-reserved-db-instances`

다음 코드 예시에서는 `describe-reserved-db-instances`을 사용하는 방법을 보여 줍니다.

AWS CLI

예약 DB 인스턴스를 설명하려면

다음 `describe-reserved-db-instances` 예제에서는 현재 AWS 계정의 예약 DB 인스턴스에 대한 세부 정보를 검색합니다.

```
aws rds describe-reserved-db-instances
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon용 예약 DB 인스턴스RDS를 참조하세요. RDS

- 자세한 API 내용은 명령 참조DescribeReservedDbInstances의 섹션을 참조하세요. AWS CLI

### `describe-source-regions`

다음 코드 예시에서는 `describe-source-regions`을 사용하는 방법을 보여 줍니다.

AWS CLI

소스 리전을 설명하려면

다음 `describe-source-regions` 예제에서는 모든 소스 AWS 리전에 대한 세부 정보를 검색합니다. 또한 자동 백업은 미국 서부(오레곤)에서 미국 동부(버지니아 북부)의 대상 AWS 리전으로만 복제할 수 있음을 보여줍니다.

```
aws rds describe-source-regions \
    --region us-east-1
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 Amazon RDS 사용 설명서의 복제된 백업에 대한 정보 찾기를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeSourceRegions의 섹션을 참조하세요. AWS CLI

### `describe-valid-db-instance-modifications`

다음 코드 예시에서는 `describe-valid-db-instance-modifications`을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 인스턴스에 대한 유효한 수정 사항을 설명하려면

다음 describe-valid-db-instance-modifications 예제에서는 지정된 DB 인스턴스의 유효한 수정에 대한 세부 정보를 검색합니다.

```
aws rds describe-valid-db-instance-modifications \
    --db-instance-identifier test-instance
```

출력:

```
{
    "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"
            }
        ]
    }
}
```

- 자세한 API 내용은 명령 참조DescribeValidDbInstanceModifications의 섹션을 참조하세요. AWS CLI

### download-db-log-file-portion

다음 코드 예시에서는 download-db-log-file-portion을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 로그 파일을 다운로드하려면

다음 download-db-log-file-portion 예제에서는 로그 파일의 최신 부분만 다운로드하여 라는 로컬 파일에 저장합니다tail.txt.

```
aws rds download-db-log-file-portion \
    --db-instance-identifier test-instance \
    --log-file-name log.txt \
    --output text > tail.txt
```

전체 파일을 다운로드하려면 --starting-token 0 파라미터를 포함해야 합니다. 다음 예제에서는 출력을 라는 로컬 파일에 저장합니다full.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
```

저장된 파일에 빈 줄이 있을 수 있습니다. 다운로드하는 동안 로그 파일의 각 부분 끝에 표시됩니다. 이는 일반적으로 로그 파일 분석에 문제를 일으키지 않습니다.

- 자세한 API 내용은 명령 참조DownloadDbLogFilePortion의 섹션을 참조하세요. AWS CLI

### generate-auth-token

다음 코드 예시에서는 generate-auth-token을 사용하는 방법을 보여 줍니다.

AWS CLI

인증 토큰을 생성하려면

다음 generate-db-auth-token 예제에서는 IAM 데이터베이스 인증에 사용할 인증 토큰을 생성합니다.

```
aws rds generate-db-auth-token \
    --hostname aurmysql-test.cdgmuqiadpid.us-west-2.rds.amazonaws.com \
    --port 3306 \
    --region us-east-1 \
    --username jane_doe
```

출력:

```
aurmysql-test.cdgmuqiadpid.us-west-2.rds.amazonaws.com:3306/?
Action=connect&DBUser=jane_doe&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Credential=AKIAIESZCNJ3OEXAMPLE%2F20180731%2Fus-east-1%2Frds-db%2Faws4_request&X-
Amz-Date=20180731T235209Z&X-Amz-Expires=900&X-Amz-SignedHeaders=host&X-Amz-
Signature=5a8753ebEXAMPLEa2c724e5667797EXAMPLE9d6ec6e3f427191fa41aeEXAMPLE
```

- 자세한 API 내용은 명령 참조GenerateAuthToken의 섹션을 참조하세요. AWS CLI

### generate-db-auth-token

다음 코드 예시에서는 generate-db-auth-token을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 인증 토큰을 생성하려면

다음 generate-db-auth-token 예제에서는 데이터베이스에 연결하기 위한 IAM 인증 토큰을 생성합니다.

```
aws rds generate-db-auth-token \
    --hostname mydb.123456789012.us-east-1.rds.amazonaws.com \
    --port 3306 \
    --region us-east-1 \
```

```
    --username db_user
```

출력:

```
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
```

자세한 내용은 Amazon RDS 사용 설명서의 IAM 인증을 사용하여 DB 인스턴스에 연결 및 Amazon Aurora 사용 설명서의 IAM 인증을 사용하여 DB 클러스터에 연결을 참조하세요.

- 자세한 API 내용은 명령 참조GenerateDbAuthToken의 섹션을 참조하세요. AWS CLI

### list-tags-for-resource

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon RDS 리소스의 태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 DB 인스턴스의 모든 태그를 나열합니다.

```
aws rds list-tags-for-resource \
    --resource-name arn:aws:rds:us-east-1:123456789012:db:orcl1
```

출력:

```
{
    "TagList": [
        {
            "Key": "Environment",
            "Value": "test"
        },
        {
            "Key": "Name",
            "Value": "MyDatabase"
        }
    ]
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon RDS 리소스 태그 지정을 참조하세요.  RDS

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### `modify-certificates`

다음 코드 예시에서는 `modify-certificates`을 사용하는 방법을 보여 줍니다.

AWS CLI

새 DB 인스턴스에 대한 시스템 기본값SSL/TLS인증서를 일시적으로 재정의하려면

다음 `modify-certificates` 예제에서는 새 DB 인스턴스에 대한 시스템 기본SSL값/TLS인증서를 일시적으로 재정의합니다.

```
aws rds modify-certificates \
    --certificate-identifier rds-ca-2019
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon RDS 사용 설명서의 SSL/TLS 인증서 교체 및 Amazon Aurora 사용 설명서의 SSL/TLS 인증서 교체를 참조하세요.

- 자세한 API 내용은 명령 참조ModifyCertificates의 섹션을 참조하세요. AWS CLI

### `modify-current-db-cluster-capacity`

다음 코드 예시에서는 `modify-current-db-cluster-capacity`을 사용하는 방법을 보여 줍니다.

AWS CLI

Aurora Serverless DB 클러스터의 용량을 조정하려면

다음 `modify-current-db-cluster-capacity` 예제에서는 Aurora Serverless DB 클러스터의 용량을 8로 조정합니다.

```
aws rds modify-current-db-cluster-capacity \
    --db-cluster-identifier mydbcluster \
    --capacity 8
```

출력:

```
{
    "DBClusterIdentifier": "mydbcluster",
    "PendingCapacity": 8,
    "CurrentCapacity": 1,
    "SecondsBeforeTimeout": 300,
    "TimeoutAction": "ForceApplyCapacityChange"
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 수동으로 Aurora Serverless v1 DB 클러스터 용량 조정을 참조하세요.

- 자세한 API 내용은 명령 참조ModifyCurrentDbClusterCapacity의 섹션을 참조하세요. AWS CLI

### `modify-db-cluster-endpoint`

다음 코드 예시에서는 `modify-db-cluster-endpoint`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 DB 클러스터 엔드포인트를 수정하려면

다음 `modify-db-cluster-endpoint` 예제에서는 지정된 사용자 지정 DB 클러스터 엔드포인트를 수정합니다.

```
aws rds modify-db-cluster-endpoint \
    --db-cluster-endpoint-identifier mycustomendpoint \
    --static-members dbinstance1 dbinstance2 dbinstance3
```

출력:

```
{
    "DBClusterEndpointIdentifier": "mycustomendpoint",
    "DBClusterIdentifier": "mydbcluster",
    "DBClusterEndpointResourceIdentifier": "cluster-endpoint-ANPAJ4AE5446DAEXAMPLE",
    "Endpoint": "mycustomendpoint.cluster-custom-cnpexample.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"
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora 연결 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ModifyDbClusterEndpoint의 섹션을 참조하세요. AWS CLI

### modify-db-cluster-parameter-group

다음 코드 예시에서는 modify-db-cluster-parameter-group을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 클러스터 파라미터 그룹의 파라미터를 수정하려면

다음 modify-db-cluster-parameter-group 예제에서는 DB 클러스터 파라미터 그룹의 파라미터 값을 수정합니다.

```
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"
```

출력:

```
{
    "DBClusterParameterGroupName": "mydbclusterpg"
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 DB 파라미터 그룹 및 DB 클러스터 파라미터 그룹 작업을 참조하세요.

- 자세한 API 내용은 명령 참조ModifyDbClusterParameterGroup의 섹션을 참조하세요. AWS CLI

### modify-db-cluster-snapshot-attribute

다음 코드 예시에서는 modify-db-cluster-snapshot-attribute을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 클러스터 스냅샷 속성을 수정하려면

다음 modify-db-cluster-snapshot-attribute 예제에서는 지정된 DB 클러스터 스냅샷 속성을 변경합니다.

```
aws rds modify-db-cluster-snapshot-attribute \
    --db-cluster-snapshot-identifier myclustersnapshot \
    --attribute-name restore \
    --values-to-add 123456789012
```

출력:

```
{
    "DBClusterSnapshotAttributesResult": {
        "DBClusterSnapshotIdentifier": "myclustersnapshot",
        "DBClusterSnapshotAttributes": [
            {
                "AttributeName": "restore",
                "AttributeValues": [
                    "123456789012"
                ]
            }
        ]
    }
```

```
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 DB 클러스터 스냅샷에서 복원을 참조하세요.

- 자세한 API 내용은 명령 참조ModifyDbClusterSnapshotAttribute의 섹션을 참조하세요. AWS CLI

### modify-db-cluster

다음 코드 예시에서는 modify-db-cluster을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: DB 클러스터 수정

다음 modify-db-cluster 예제에서는 라는 DB 클러스터의 마스터 사용자 암호를 변경cluster-2하고 백업 보존 기간을 14일로 설정합니다. --apply-immediately 파라미터는 다음 유지 관리 기간까지 기다리는 대신 즉시 변경 사항을 적용합니다.

```
aws rds modify-db-cluster \
    --db-cluster-identifier cluster-2 \
    --backup-retention-period 14 \
    --master-user-password newpassword99 \
    --apply-immediately
```

출력:

```
{
    "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": []
    }
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 수정을 참조하세요.

예제 2: VPC 보안 그룹을 DB 클러스터에 연결하려면

다음 modify-db-instance 예제는 특정 VPC 보안 그룹을 연결하고 DB 클러스터에서 DB 보안 그룹을 제거합니다.

```
aws rds modify-db-cluster \
    --db-cluster-identifier dbName \
    --vpc-security-group-ids sg-ID
```

출력:

```
{
    "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...
    }
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 보안 그룹을 사용하여 액세스 제어를 참조하세요.

- 자세한 API 내용은 명령 참조ModifyDbCluster의 섹션을 참조하세요. AWS CLI

### `modify-db-instance`

다음 코드 예시에서는 `modify-db-instance`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: DB 인스턴스를 수정하려면 다음과 같이 하세요.

다음 `modify-db-instance` 예제에서는 옵션 그룹과 파라미터 그룹을 호환되는 Microsoft SQL Server DB 인스턴스와 연결합니다. `--apply-immediately` 파라미터를 사용하면 다음 유지 관리 기간이 될 때까지 기다리는 대신 옵션과 파라미터 그룹이 즉시 연결됩니다.

```
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
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon RDS DB 인스턴스 수정을 참조하세요. RDS

예제 2: VPC 보안 그룹을 DB 인스턴스와 연결하려면

다음 `modify-db-instance` 예제는 특정 VPC 보안 그룹을 연결하고 DB 인스턴스에서 DB 보안 그룹을 제거합니다.

```
aws rds modify-db-instance \
    --db-instance-identifier dbName \
    --vpc-security-group-ids sg-ID
```

출력:

```
{
"DBInstance": {
    "DBInstanceIdentifier": "dbName",
    "DBInstanceClass": "db.t3.micro",
    "Engine": "mysql",
    "DBInstanceStatus": "available",
    "MasterUsername": "admin",
    "Endpoint": {
        "Address": "dbName.abcdefghijk.us-west-2.rds.amazonaws.com",
        "Port": 3306,
        "HostedZoneId": "ABCDEFGHIJK1234"
    },
    "AllocatedStorage": 20,
    "InstanceCreateTime": "2024-02-15T00:37:58.793000+00:00",
    "PreferredBackupWindow": "11:57-12:27",
    "BackupRetentionPeriod": 7,
    "DBSecurityGroups": [],
    "VpcSecurityGroups": [
        {
            "VpcSecurityGroupId": "sg-ID",
            "Status": "active"
        }
    ],
    ... output omitted ...
    "MultiAZ": false,
    "EngineVersion": "8.0.35",
    "AutoMinorVersionUpgrade": true,
    "ReadReplicaDBInstanceIdentifiers": [],
    "LicenseModel": "general-public-license",

    ... output ommited ...
    }
}
```

자세한 내용은 Amazon RDS 사용 설명서의 보안 그룹을 사용하여 액세스 제어를 참조하세요.

- API 자세한 내용은 AWS CLI 명령 참조의 ModifyDBInstance을 참조하세요.

### modify-db-parameter-group

다음 코드 예시에서는 modify-db-parameter-group을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 파라미터 그룹을 수정하려면

다음 modify-db-parameter-group 예제에서는 DB 파라미터 그룹의 clr enabled 파라미터 값을 변경합니다. --apply-immediately 파라미터를 사용하면 다음 유지 관리 기간이 될 때까지 기다리는 대신 DB 파라미터 그룹이 즉시 수정됩니다.

```
aws rds modify-db-parameter-group \
    --db-parameter-group-name test-sqlserver-se-2017 \
    --parameters "ParameterName='clr
 enabled',ParameterValue=1,ApplyMethod=immediate"
```

출력:

```
{
    "DBParameterGroupName": "test-sqlserver-se-2017"
}
```

자세한 내용은 Amazon RDS 사용 설명서의 DB 파라미터 그룹에서 파라미터 수정을 참조하세요.

- API 자세한 내용은 명령 참조의 M odifyDBParameter그룹을 참조하세요. AWS CLI

### modify-db-proxy-endpoint

다음 코드 예시에서는 modify-db-proxy-endpoint을 사용하는 방법을 보여 줍니다.

AWS CLI

RDS 데이터베이스에 대한 DB 프록시 엔드포인트를 수정하려면

다음 modify-db-proxy-endpoint 예제에서는 DB 프록시 엔드포인트를 수정proxyEndpoint하여 읽기 제한 시간을 65초로 설정합니다.

```
aws rds modify-db-proxy-endpoint \
    --db-proxy-endpoint-name proxyEndpoint \
    --cli-read-timeout 65
```

출력:

```
{
"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",
        "CreatedDate": "2023-04-05T16:09:33.452000+00:00",
        "TargetRole": "READ_WRITE",
        "IsDefault": "false"
    }
}
```

자세한 내용은 Amazon RDS 사용 설명서의 프록시 엔드포인트 수정 및 Amazon Aurora 사용 설명서의 프록시 엔드포인트 수정을 참조하세요.

- 자세한 API 내용은 명령 참조ModifyDbProxyEndpoint의 섹션을 참조하세요. AWS CLI

### modify-db-proxy-target-group

다음 코드 예시에서는 modify-db-proxy-target-group을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 프록시 엔드포인트를 수정하려면

다음 modify-db-proxy-target-group 예제에서는 DB 프록시 대상 그룹을 수정하여 최대 연결을 80%로 설정하고 최대 유휴 연결을 10%로 설정합니다.

```
aws rds modify-db-proxy-target-group \
```

```
    --target-group-name default \
    --db-proxy-name proxyExample \
    --connection-pool-config MaxConnectionsPercent=80,MaxIdleConnectionsPercent=10
```

출력:

```
{
"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": []
        },
        "CreatedDate": "2023-05-02T18:41:19.495000+00:00",
        "UpdatedDate": "2023-05-02T18:41:21.762000+00:00"
    }
}
```

자세한 내용은 Amazon RDS 사용 설명서의 RDS 프록시 수정 및 Amazon Aurora 사용 설명서의 RDS 프록시 수정을 참조하세요.

- 자세한 API 내용은 명령 참조ModifyDbProxyTargetGroup의 섹션을 참조하세요. AWS CLI

### modify-db-proxy

다음 코드 예시에서는 modify-db-proxy을 사용하는 방법을 보여 줍니다.

AWS CLI

RDS 데이터베이스의 DB 프록시를 수정하려면

다음 modify-db-proxy 예제에서는 연결SSL에 가 필요하다고 명명된 DB 프록시proxyExample를 수정합니다.

```
aws rds modify-db-proxy \
```

```
    --db-proxy-name proxyExample \
    --require-tls
```

출력:

```
{
"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,
        "CreatedDate": "2023-04-05T16:09:33.452000+00:00",
        "UpdatedDate": "2023-04-13T01:49:38.568000+00:00"
    }
}
```

자세한 내용은 Amazon RDS 사용 설명서의 RDS 프록시 수정 및 Amazon Aurora 사용 설명서의 RDS 프록시 생성을 참조하세요.

- 자세한 API 내용은 명령 참조ModifyDbProxy의 섹션을 참조하세요. AWS CLI

### `modify-db-shard-group`

다음 코드 예시에서는 `modify-db-shard-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: DB 샤드 그룹을 수정하려면

다음 `modify-db-shard-group` 예제에서는 DB 샤드 그룹의 최대 용량을 변경합니다.

```
aws rds modify-db-shard-group \
    --db-shard-group-identifier my-db-shard-group \
    --max-acu 1000
```

출력:

```
{
    "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-cekycexample.us-
east-2.rds.amazonaws.com"
        }
    ]
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터를 참조하세요.

예제 2: DB 샤드 그룹 설명

다음 `describe-db-shard-groups` 예제에서는 `modify-db-shard-group` 명령을 실행한 후 DB 샤드 그룹의 세부 정보를 검색합니다. DB 샤드 그룹의 최대 용량은 이제 1000 Aurora 용량 단위()`my-db-shard-group`입니다ACUs.

```
aws rds describe-db-shard-groups
```

출력:

```
{
    "DBShardGroups": [
        {
            "DBShardGroupResourceId": "shardgroup-7bb446329da94788b3f957746example",
            "DBShardGroupIdentifier": "limitless-test-shard-grp",
            "DBClusterIdentifier": "limitless-test-cluster",
            "MaxACU": 768.0,
            "ComputeRedundancy": 0,
            "Status": "available",
            "PubliclyAccessible": true,
            "Endpoint": "limitless-test-cluster.limitless-cekycexample.us-
east-2.rds.amazonaws.com"
        },
        {
            "DBShardGroupResourceId": "shardgroup-a6e3a0226aa243e2ac6c7a1234567890",
            "DBShardGroupIdentifier": "my-db-shard-group",
            "DBClusterIdentifier": "my-sv2-cluster",
            "MaxACU": 1000.0,
            "ComputeRedundancy": 0,
            "Status": "available",
            "PubliclyAccessible": false,
            "Endpoint": "my-sv2-cluster.limitless-cekycexample.us-
east-2.rds.amazonaws.com"
        }
    ]
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터를 참조하세요.

- 자세한 API 내용은 명령 참조ModifyDbShardGroup의 섹션을 참조하세요. AWS CLI

### `modify-db-snapshot-attribute`

다음 코드 예시에서는 `modify-db-snapshot-attribute`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 두 AWS 계정이 DB 스냅샷을 복원하도록 활성화하려면

다음 modify-db-snapshot-attribute 예제에서는 식별자 111122223333 및 를 사용하여 라는 DB 스냅샷을 복원444455556666할 수 있는 권한을 두 AWS 계정에 부여합니다mydbsnapshot.

```
aws rds modify-db-snapshot-attribute \
    --db-snapshot-identifier mydbsnapshot \
    --attribute-name restore \
    --values-to-add {"111122223333","444455556666"}
```

출력:

```
{
    "DBSnapshotAttributesResult": {
        "DBSnapshotIdentifier": "mydbsnapshot",
        "DBSnapshotAttributes": [
            {
                "AttributeName": "restore",
                "AttributeValues": [
                    "111122223333",
                    "444455556666"
                ]
            }
        ]
    }
}
```

자세한 내용은 Amazon RDS 사용 설명서의 스냅샷 공유를 참조하세요.

예제 2: AWS 계정이 DB 스냅샷을 복원하지 못하도록 하려면

다음 modify-db-snapshot-attribute 예제에서는 라는 DB 스냅샷을 복원할 수 있는 권한을 특정 AWS 계정에서 제거합니다mydbsnapshot. 단일 계정을 지정할 때 계정 식별자는 따옴표 또는 부호로 둘러쌀 수 없습니다.

```
aws rds modify-db-snapshot-attribute \
    --db-snapshot-identifier mydbsnapshot \
    --attribute-name restore \
    --values-to-remove 444455556666
```

출력:

```
{
```

```
    "DBSnapshotAttributesResult": {
        "DBSnapshotIdentifier": "mydbsnapshot",
        "DBSnapshotAttributes": [
            {
                "AttributeName": "restore",
                "AttributeValues": [
                    "111122223333"
                ]
            }
        ]
    }
}
```

자세한 내용은 Amazon RDS 사용 설명서의 스냅샷 공유를 참조하세요.

- 자세한 API 내용은 명령 참조ModifyDbSnapshotAttribute의 섹션을 참조하세요. AWS CLI

### `modify-db-snapshot-attributes`

다음 코드 예시에서는 `modify-db-snapshot-attributes`을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 스냅샷 속성을 수정하려면

다음 `modify-db-snapshot-attribute` 예제에서는 라는 DB 스냅샷을 복원444455556666하기 위해 두 개의 AWS 계정 식별자 111122223333 및 를 허용합니다mydbsnapshot.

```
aws rds modify-db-snapshot-attribute \
    --db-snapshot-identifier mydbsnapshot \
    --attribute-name restore \
    --values-to-add '["111122223333","444455556666"]'
```

출력:

```
{
    "DBSnapshotAttributesResult": {
        "DBSnapshotIdentifier": "mydbsnapshot",
        "DBSnapshotAttributes": [
            {
                "AttributeName": "restore",
                "AttributeValues": [
                    "111122223333",
```

```
                    "444455556666"
                ]
            }
        ]
    }
}
```

자세한 내용은 Amazon RDS 사용 설명서의 스냅샷 공유를 참조하세요.

- 자세한 API 내용은 명령 참조ModifyDbSnapshotAttributes의 섹션을 참조하세요. AWS CLI

### modify-db-snapshot

다음 코드 예시에서는 modify-db-snapshot을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 스냅샷을 수정하려면

다음 modify-db-snapshot 예제에서는 라는 PostgeSQL 10.6 스냅샷을 PostgreSQL 11.7db5-snapshot-upg-test로 업그레이드합니다. 스냅샷 업그레이드가 완료되고 상태를 사용할 수 있게 되면 새 DB 엔진 버전이 표시됩니다.

```
aws rds modify-db-snapshot \
    --db-snapshot-identifier db5-snapshot-upg-test \
    --engine-version 11.7
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon RDS 사용 설명서의 PostgreSQL DB 스냅샷 업그레이드를 참조하세요.

- 자세한 API 내용은 명령 참조ModifyDbSnapshot의 섹션을 참조하세요. AWS CLI

### modify-db-subnet-group

다음 코드 예시에서는 modify-db-subnet-group을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 서브넷 그룹을 수정하려면

다음 modify-db-subnet-group 예제에서는 ID가 인 서브넷subnet-08e41f9e230222222을 라는 DB 서브넷 그룹에 추가합니다mysubnetgroup. 서브넷 그룹에 기존 서브넷을 유지하려면 --subnet-ids 옵션에 IDs 해당 서브넷을 값으로 포함시킵니다. DB 서브넷 그룹에 최소 두 개의 서로 다른 가용 영역이 있는 서브넷이 있어야 합니다.

```
aws rds modify-db-subnet-group \
    --db-subnet-group-name mysubnetgroup \
    --subnet-ids
 '["subnet-0a1dc4e1a6f123456","subnet-070dd7ecb3aaaaaaa","subnet-00f5b198bc0abcdef","subnet-
```

출력:

```
{
    "DBSubnetGroup": {
        "DBSubnetGroupName": "mysubnetgroup",
        "DBSubnetGroupDescription": "test DB subnet group",
        "VpcId": "vpc-0f08e7610a1b2c3d4",
```

```
        "SubnetGroupStatus": "Complete",
        "Subnets": [
            {
                "SubnetIdentifier": "subnet-08e41f9e230222222",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2a"
                },
                "SubnetStatus": "Active"
            },
            {
                "SubnetIdentifier": "subnet-070dd7ecb3aaaaaaa",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2b"
                },
                "SubnetStatus": "Active"
            },
            {
                "SubnetIdentifier": "subnet-00f5b198bc0abcdef",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2d"
                },
                "SubnetStatus": "Active"
            },
            {
                "SubnetIdentifier": "subnet-0a1dc4e1a6f123456",
                "SubnetAvailabilityZone": {
                    "Name": "us-west-2b"
                },
                "SubnetStatus": "Active"
            }
        ],
        "DBSubnetGroupArn": "arn:aws:rds:us-
west-2:534026745191:subgrp:mysubnetgroup"
    }
}
```

자세한 내용은 Amazon RDS 사용 설명서의 3단계: DB 서브넷 그룹 생성을 참조하세요.

- 자세한 API 내용은 명령 참조ModifyDbSubnetGroup의 섹션을 참조하세요. AWS CLI

### `modify-event-subscription`

다음 코드 예시에서는 `modify-event-subscription`을 사용하는 방법을 보여 줍니다.

AWS CLI

이벤트 구독을 수정하려면

다음 modify-event-subscription 예제에서는 지정된 이벤트 구독을 비활성화하므로 지정된 Amazon Simple Notification Service 주제에 더 이상 알림을 게시하지 않습니다.

```
aws rds modify-event-subscription \
    --subscription-name my-instance-events \
    --no-enabled
```

출력:

```
{
    "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
    }
}
```

- 자세한 API 내용은 명령 참조ModifyEventSubscription의 섹션을 참조하세요. AWS CLI

### modify-global-cluster

다음 코드 예시에서는 modify-global-cluster을 사용하는 방법을 보여 줍니다.

AWS CLI

글로벌 DB 클러스터를 수정하려면

다음 modify-global-cluster 예제에서는 Aurora My SQL호환 글로벌 DB 클러스터에 대한 삭제 보호를 활성화합니다.

```
aws rds modify-global-cluster \
    --global-cluster-identifier myglobalcluster \
    --deletion-protection
```

출력:

```
{
    "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": []
    }
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 Aurora 글로벌 데이터베이스 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ModifyGlobalCluster의 섹션을 참조하세요. AWS CLI

### promote-read-replica-db-cluster

다음 코드 예시에서는 promote-read-replica-db-cluster을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 클러스터 읽기 전용 복제본을 승격하려면

다음 promote-read-replica-db-cluster 예제에서는 지정된 읽기 전용 복제본을 독립 실행형 DB 클러스터로 승격합니다.

```
aws rds promote-read-replica-db-cluster \
    --db-cluster-identifier mydbcluster-1
```

출력:

```
{
    "DBCluster": {
        "AllocatedStorage": 1,
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1b",
            "us-east-1c"
        ],
        "BackupRetentionPeriod": 1,
        "DatabaseName": "",
        "DBClusterIdentifier": "mydbcluster-1",
        ...some output truncated...
    }
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 읽기 전용 복제본을 DB 클러스터로 승격을 참조하세요.

- 자세한 API 내용은 명령 참조PromoteReadReplicaDbCluster의 섹션을 참조하세요. AWS CLI

### promote-read-replica

다음 코드 예시에서는 promote-read-replica을 사용하는 방법을 보여 줍니다.

AWS CLI

읽기 전용 복제본을 승격하려면

다음 promote-read-replica 예제에서는 지정된 읽기 전용 복제본을 독립 실행형 DB 인스턴스로 승격합니다.

```
aws rds promote-read-replica \
    --db-instance-identifier test-instance-repl
```

출력:

```
{
    "DBInstance": {
        "DBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:test-instance-repl",
        "StorageType": "standard",
```

```
        "ReadReplicaSourceDBInstanceIdentifier": "test-instance",
        "DBInstanceStatus": "modifying",
        ...some output truncated...
    }
}
```

- 자세한 API 내용은 명령 참조PromoteReadReplica의 섹션을 참조하세요. AWS CLI

### purchase-reserved-db-instance

다음 코드 예시에서는 purchase-reserved-db-instance을 사용하는 방법을 보여 줍니다.

AWS CLI

예약 DB 인스턴스 제품을 구매하려면

다음 purchase-reserved-db-instances-offering 예제에서는 예약 DB 인스턴스 제안을 구매합니다. 는 describe-reserved-db-instances-offering 명령으로 반환된 유효한 제공 ID여야 reserved-db-instances-offering-id 합니다.

aws rds purchase-reserved-db-instances-offering -reserved-db-instances-offering-id 438012d3-4a52-4cc7-b2e3-8dff72e0e706

- 자세한 API 내용은 명령 참조PurchaseReservedDbInstance의 섹션을 참조하세요. AWS CLI

### purchase-reserved-db-instances-offerings

다음 코드 예시에서는 purchase-reserved-db-instances-offerings을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 구매할 예약 DB 인스턴스를 찾으려면

다음 describe-reserved-db-instances-offerings 예제에서는 db.t2.micro 인스턴스 클래스와 1년의 기간이 있는 사용 가능한 예약 MySQL DB 인스턴스를 나열합니다. 예약 DB 인스턴스를 구매하려면 제공 ID가 필요합니다.

```
aws rds describe-reserved-db-instances-offerings \
    --product-description mysql \
    --db-instance-class db.t2.micro \
```

```
    --duration 1
```

출력:

```
{
    "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 ...
    ]
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon용 예약 DB 인스턴스RDS를 참조하세요. RDS

예제 2: 예약 DB 인스턴스를 구매하려면

다음 `purchase-reserved-db-instances-offering` 예제에서는 이전 예제에서 예약 DB 인스턴스 제품을 구매하는 방법을 보여줍니다.

aws rds purchase-reserved-db-instances-offering -reserved-db-instances-offering-id 8ba30be1-b9ec-447f-8f23-6114e3f4c7b4

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon용 예약 DB 인스턴스RDS를 참조하세요. RDS

- 자세한 API 내용은 명령 참조PurchaseReservedDbInstancesOfferings의 섹션을 참조하세요. AWS CLI

### reboot-db-instance

다음 코드 예시에서는 reboot-db-instance을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 인스턴스를 재부팅하려면

다음 reboot-db-instance 예제에서는 지정된 DB 인스턴스의 재부팅을 시작합니다.

```
aws rds reboot-db-instance \
    --db-instance-identifier test-mysql-instance
```

출력:

```
{
    "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...

    }
}
```

자세한 내용은 Amazon RDS 사용 설명서의 DB 인스턴스 재부팅을 참조하세요.

- API 자세한 내용은 AWS CLI 명령 참조의 RebootDBInstance을 참조하세요.

### reboot-db-shard-group

다음 코드 예시에서는 reboot-db-shard-group을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: DB 샤드 그룹을 재부팅하려면

다음 reboot-db-shard-group 예제에서는 DB 샤드 그룹을 재부팅합니다.

```
aws rds reboot-db-shard-group \
    --db-shard-group-identifier my-db-shard-group
```

출력:

```
{
    "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-cekycexample.us-
east-2.rds.amazonaws.com"
        }
    ]
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 또는 Amazon Aurora DB 인스턴스 재부팅을 참조하세요.

예제 2: DB 샤드 그룹을 설명하려면

다음 `describe-db-shard-groups` 예제에서는 `reboot-db-shard-group` 명령을 실행한 후 DB 샤드 그룹의 세부 정보를 검색합니다. `my-db-shard-group` 이제 DB 샤드 그룹이 재부팅 중입니다.

```
aws rds describe-db-shard-groups
```

출력:

```
{
    "DBShardGroups": [
        {
            "DBShardGroupResourceId": "shardgroup-7bb446329da94788b3f957746example",
            "DBShardGroupIdentifier": "limitless-test-shard-grp",
            "DBClusterIdentifier": "limitless-test-cluster",
            "MaxACU": 768.0,
            "ComputeRedundancy": 0,
            "Status": "available",
            "PubliclyAccessible": true,
            "Endpoint": "limitless-test-cluster.limitless-cekycexample.us-
east-2.rds.amazonaws.com"
        },
        {
            "DBShardGroupResourceId": "shardgroup-a6e3a0226aa243e2ac6c7a1234567890",
            "DBShardGroupIdentifier": "my-db-shard-group",
            "DBClusterIdentifier": "my-sv2-cluster",
```

```
            "MaxACU": 1000.0,
            "ComputeRedundancy": 0,
            "Status": "rebooting",
            "PubliclyAccessible": false,
            "Endpoint": "my-sv2-cluster.limitless-cekycexample.us-
east-2.rds.amazonaws.com"
        }
    ]
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 또는 Amazon Aurora DB 인스턴스 재부팅을 참조하세요.

- 자세한 API 내용은 명령 참조RebootDbShardGroup의 섹션을 참조하세요. AWS CLI

### register-db-proxy-targets

다음 코드 예시에서는 register-db-proxy-targets을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터베이스에 DB 프록시를 등록하려면

다음 register-db-proxy-targets 예제에서는 데이터베이스와 프록시 간의 연결을 생성합니다.

```
aws rds register-db-proxy-targets \
    --db-proxy-name proxyExample \
    --db-cluster-identifiers database-5
```

출력:

```
{
    "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"
            }
        }
    ]
}
```

자세한 내용은 Amazon RDS 사용 설명서의 RDS 프록시 생성 및 Amazon Aurora 사용 설명서의 RDS 프록시 생성을 참조하세요.

- 자세한 API 내용은 명령 참조RegisterDbProxyTargets의 섹션을 참조하세요. AWS CLI

### `remove-from-global-cluster`

다음 코드 예시에서는 `remove-from-global-cluster`을 사용하는 방법을 보여 줍니다.

AWS CLI

Aurora 보조 클러스터를 Aurora 글로벌 데이터베이스 클러스터에서 분리하려면

다음 `remove-from-global-cluster` 예제에서는 Aurora 보조 클러스터를 Aurora 글로벌 데이터베이스 클러스터에서 분리합니다. 클러스터는 읽기 전용에서 읽기-쓰기 기능이 있는 독립 실행형 클러스터로 변경됩니다.

```
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
```

출력:

```
{
    "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"
            }
        ]
    }
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora 글로벌 데이터베이스에서 클러스터 제거를 참조하세요.

- 자세한 API 내용은 명령 참조RemoveFromGlobalCluster의 섹션을 참조하세요. AWS CLI

### remove-option-from-option-group

다음 코드 예시에서는 remove-option-from-option-group을 사용하는 방법을 보여 줍니다.

AWS CLI

옵션 그룹에서 옵션을 삭제하려면

다음 remove-option-from-option-group 예제에서는 에서 OEM 옵션을 제거합니다myoptiongroup.

```
aws rds remove-option-from-option-group \
```

```
    --option-group-name myoptiongroup \
    --options OEM \
    --apply-immediately
```

출력:

```
{
    "OptionGroup": {
        "OptionGroupName": "myoptiongroup",
        "OptionGroupDescription": "Test",
        "EngineName": "oracle-ee",
        "MajorEngineVersion": "19",
        "Options": [],
        "AllowsVpcAndNonVpcInstanceMemberships": true,
        "OptionGroupArn": "arn:aws:rds:us-east-1:123456789012:og:myoptiongroup"
    }
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 옵션 그룹에서 옵션 제거를 참조하세요.

- 자세한 API 내용은 명령 참조RemoveOptionFromOptionGroup의 섹션을 참조하세요. AWS CLI

### `remove-role-from-db-cluster`

다음 코드 예시에서는 `remove-role-from-db-cluster`을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 클러스터에서 AWS 자격 증명 및 액세스 관리(IAM) 역할을 연결 해제하려면

다음 `remove-role-from-db-cluster` 예제에서는 DB 클러스터에서 역할을 제거합니다.

```
aws rds remove-role-from-db-cluster \
    --db-cluster-identifier mydbcluster \
    --role-arn arn:aws:iam::123456789012:role/RDSLoadFromS3
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora MySQL DB 클러스터와 IAM 역할 연결을 참조하세요.

- 자세한 API 내용은 명령 참조RemoveRoleFromDbCluster의 섹션을 참조하세요. AWS CLI

### remove-role-from-db-instance

다음 코드 예시에서는 remove-role-from-db-instance을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 인스턴스에서 AWS 자격 증명 및 액세스 관리(IAM) 역할을 연결 해제하려면

다음 remove-role-from-db-instance 예제에서는 라는 Oracle DB 인스턴스rds-s3-integration-role에서 라는 역할을 제거합니다test-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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon RDS 사용 설명서의 S3와 RDS SQL 서버 통합 비활성화를 참조하세요.

- 자세한 API 내용은 명령 참조RemoveRoleFromDbInstance의 섹션을 참조하세요. AWS CLI

### remove-source-identifier-from-subscription

다음 코드 예시에서는 remove-source-identifier-from-subscription을 사용하는 방법을 보여 줍니다.

AWS CLI

구독에서 소스 식별자를 제거하려면

다음 remove-source-identifier 예제에서는 기존 구독에서 지정된 소스 식별자를 제거합니다.

```
aws rds remove-source-identifier-from-subscription \
    --subscription-name my-instance-events \
    --source-identifier test-instance-repl
```

출력:

```
{
    "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
    }
}
```

- 자세한 API 내용은 명령 참조RemoveSourceIdentifierFromSubscription의 섹션을 참조하세요. AWS CLI

### `remove-tags-from-resource`

다음 코드 예시에서는 `remove-tags-from-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에서 태그를 제거하려면

다음 `remove-tags-from-resource` 예제에서는 리소스에서 태그를 제거합니다.

```
aws rds remove-tags-from-resource \
    --resource-name arn:aws:rds:us-east-1:123456789012:db:mydbinstance \
    --tag-keys Name Environment
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon 사용 설명서의 Amazon RDS 리소스 태그 지정 및 Amazon Aurora 사용 설명서의 Amazon RDS 리소스 태그를 참조하세요. RDS

- 자세한 API 내용은 명령 참조RemoveTagsFromResource의 섹션을 참조하세요. AWS CLI

### reset-db-cluster-parameter-group

다음 코드 예시에서는 reset-db-cluster-parameter-group을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 모든 파라미터를 기본값으로 재설정하려면

다음 reset-db-cluster-parameter-group 예제에서는 고객이 생성한 DB 클러스터 파라미터 그룹의 모든 파라미터 값을 기본값으로 재설정합니다.

```
aws rds reset-db-cluster-parameter-group \
    --db-cluster-parameter-group-name mydbclpg \
    --reset-all-parameters
```

출력:

```
{
    "DBClusterParameterGroupName": "mydbclpg"
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 DB 파라미터 그룹 및 DB 클러스터 파라미터 그룹 작업을 참조하세요.

예제 2: 특정 파라미터를 기본값으로 재설정하려면

다음 reset-db-cluster-parameter-group 예제에서는 특정 파라미터의 파라미터 값을 고객이 생성한 DB 클러스터 파라미터 그룹의 기본값으로 재설정합니다.

```
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"
```

출력:

```
{
    "DBClusterParameterGroupName": "mydbclpg"
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 DB 파라미터 그룹 및 DB 클러스터 파라미터 그룹 작업을 참조하세요.

- 자세한 API 내용은 명령 참조ResetDbClusterParameterGroup의 섹션을 참조하세요. AWS CLI

### reset-db-parameter-group

다음 코드 예시에서는 reset-db-parameter-group을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 모든 파라미터를 기본값으로 재설정하려면

다음 reset-db-parameter-group 예제에서는 고객이 생성한 DB 파라미터 그룹의 모든 파라미터 값을 기본값으로 재설정합니다.

```
aws rds reset-db-parameter-group \
    --db-parameter-group-name mypg \
    --reset-all-parameters
```

출력:

```
{
    "DBParameterGroupName": "mypg"
}
```

자세한 내용은 Amazon RDS 사용 설명서의 DB 파라미터 그룹 작업 및 Amazon Aurora 사용 설명서의 DB 파라미터 그룹 및 DB 클러스터 파라미터 그룹 작업을 참조하세요.

예제 2: 특정 파라미터를 기본값으로 재설정하려면

다음 reset-db-parameter-group 예제에서는 특정 파라미터의 파라미터 값을 고객이 생성한 DB 파라미터 그룹의 기본값으로 재설정합니다.

```
aws rds reset-db-parameter-group \
    --db-parameter-group-name mypg \
    --parameters "ParameterName=max_connections,ApplyMethod=immediate" \
                 "ParameterName=max_allowed_packet,ApplyMethod=immediate"
```

출력:

```
{
    "DBParameterGroupName": "mypg"
}
```

자세한 내용은 Amazon RDS 사용 설명서의 DB 파라미터 그룹 작업 및 Amazon Aurora 사용 설명서의 DB 파라미터 그룹 및 DB 클러스터 파라미터 그룹 작업을 참조하세요.

- 자세한 API 내용은 명령 참조ResetDbParameterGroup의 섹션을 참조하세요. AWS CLI

### `restore-db-cluster-from-s3`

다음 코드 예시에서는 `restore-db-cluster-from-s3`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon S3에서 Amazon Aurora DB 클러스터를 복원하려면

다음 `restore-db-cluster-from-s3` 예제는 Amazon S3의 MySQL 5.7 DB 백업 파일에서 Amazon Aurora MySQL 버전 5.7 호환 DB 클러스터를 복원합니다.

```
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
```

출력:

```
{
    "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-SU5THYQQHOWCXZZDGXREXAMPLE",
        "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": []
    }
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 Amazon S3 버킷을 사용하여 내SQL 데이터 마이그레이션을 참조하세요.

- API 자세한 내용은 AWS CLI 명령 참조의 RestoreDbClusterFromS3를 참조하세요.

### restore-db-cluster-from-snapshot

다음 코드 예시에서는 restore-db-cluster-from-snapshot을 사용하는 방법을 보여 줍니다.

AWS CLI

스냅샷에서 DB 클러스터를 복원하려면

다음은 라는 DB 클러스터 스냅샷에서 PostgreSQL 버전 10.7과 호환되는 Aurora PostgreSQL DB 클러스터를 restore-db-cluster-from-snapshot 복원합니다test-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
```

출력:

```
{
    "DBCluster": {
        "AllocatedStorage": 1,
        "AvailabilityZones": [
            "us-west-2c",
            "us-west-2a",
            "us-west-2b"
        ],
        "BackupRetentionPeriod": 7,
        "DatabaseName": "",
        "DBClusterIdentifier": "newdbcluster",
        "DBClusterParameterGroup": "default.aurora-postgresql10",
        "DBSubnetGroup": "default",
        "Status": "creating",
        "Endpoint": "newdbcluster.cluster-############.us-west-2.rds.amazonaws.com",
        "ReaderEndpoint": "newdbcluster.cluster-ro-############.us-
west-2.rds.amazonaws.com",
        "MultiAZ": false,
        "Engine": "aurora-postgresql",
        "EngineVersion": "10.7",
        "Port": 5432,
        "MasterUsername": "postgres",
        "PreferredBackupWindow": "09:33-10:03",
        "PreferredMaintenanceWindow": "sun:12:22-sun:12:52",
```

```
        "ReadReplicaIdentifiers": [],
        "DBClusterMembers": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-########",
                "Status": "active"
            }
        ],
        "HostedZoneId": "Z1PVIF0EXAMPLE",
        "StorageEncrypted": true,
        "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/287364e4-33e3-4755-a3b0-
a1b2c3d4e5f6",
        "DbClusterResourceId": "cluster-5DSB5IFQDDUVAWOUWM1EXAMPLE",
        "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:newdbcluster",
        "AssociatedRoles": [],
        "IAMDatabaseAuthenticationEnabled": false,
        "ClusterCreateTime": "2020-06-05T15:06:58.634Z",
        "EngineMode": "provisioned",
        "DeletionProtection": false,
        "HttpEndpointEnabled": false,
        "CopyTagsToSnapshot": false,
        "CrossAccountClone": false,
        "DomainMemberships": []
    }
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 DB 클러스터 스냅샷에서 복원을 참조하세요.

- 자세한 API 내용은 명령 참조RestoreDbClusterFromSnapshot의 섹션을 참조하세요. AWS CLI

### `restore-db-cluster-to-point-in-time`

다음 코드 예시에서는 `restore-db-cluster-to-point-in-time`을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 클러스터를 지정된 시간으로 복원하려면

다음 `restore-db-cluster-to-point-in-time` 예제에서는 라는 DB 클러스터`database-4`를 가능한 가장 늦은 시간으로 복원합니다. 를 복원 유형`copy-on-write`으로 사용하면 새 DB 클러스터가 소스 DB 클러스터의 복제본으로 복원됩니다.

```
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
```

출력:

```
{
    "DBCluster": {
        "AllocatedStorage": 1,
        "AvailabilityZones": [
            "us-west-2c",
            "us-west-2a",
            "us-west-2b"
        ],
        "BackupRetentionPeriod": 7,
        "DatabaseName": "",
        "DBClusterIdentifier": "sample-cluster-clone",
        "DBClusterParameterGroup": "default.aurora-postgresql10",
        "DBSubnetGroup": "default",
        "Status": "creating",
        "Endpoint": "sample-cluster-clone.cluster-############.us-
west-2.rds.amazonaws.com",
        "ReaderEndpoint": "sample-cluster-clone.cluster-ro-############.us-
west-2.rds.amazonaws.com",
        "MultiAZ": false,
        "Engine": "aurora-postgresql",
        "EngineVersion": "10.7",
        "Port": 5432,
        "MasterUsername": "postgres",
        "PreferredBackupWindow": "09:33-10:03",
        "PreferredMaintenanceWindow": "sun:12:22-sun:12:52",
        "ReadReplicaIdentifiers": [],
        "DBClusterMembers": [],
        "VpcSecurityGroups": [
            {
                "VpcSecurityGroupId": "sg-########",
                "Status": "active"
            }
        ],
        "HostedZoneId": "Z1PVIF0EXAMPLE",
        "StorageEncrypted": true,
```

```
        "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/287364e4-33e3-4755-a3b0-
a1b2c3d4e5f6",
        "DbClusterResourceId": "cluster-BIZ77GDSA2XBSTNPFW1EXAMPLE",
        "DBClusterArn": "arn:aws:rds:us-west-2:123456789012:cluster:sample-cluster-
clone",
        "AssociatedRoles": [],
        "IAMDatabaseAuthenticationEnabled": false,
        "CloneGroupId": "8d19331a-099a-45a4-b4aa-11aa22bb33cc44dd",
        "ClusterCreateTime": "2020-03-10T19:57:38.967Z",
        "EngineMode": "provisioned",
        "DeletionProtection": false,
        "HttpEndpointEnabled": false,
        "CopyTagsToSnapshot": false,
        "CrossAccountClone": false
    }
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 DB 클러스터를 지정된 시간으로 복원을 참조하세요.

- 자세한 API 내용은 명령 참조RestoreDbClusterToPointInTime의 섹션을 참조하세요. AWS CLI

### `restore-db-instance-from-db-snapshot`

다음 코드 예시에서는 `restore-db-instance-from-db-snapshot`을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 스냅샷에서 DB 인스턴스를 복원하려면

다음 `restore-db-instance-from-db-snapshot` 예제에서는 지정된 DB 스냅샷에서 `db.t3.small` DB 인스턴스 클래스`db7-new-instance`로 이름이 지정된 새 DB 인스턴스를 생성합니다. 스냅샷이 생성된 소스 DB 인스턴스는 더 이상 사용되지 않는 DB 인스턴스 클래스를 사용하므로 업그레이드할 수 없습니다.

```
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
```

출력:

```
{
    "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": []
    }
}
```

자세한 내용은 Amazon RDS 사용 설명서의 DB 스냅샷에서 복원을 참조하세요.

- 자세한 API 내용은 명령 참조RestoreDbInstanceFromDbSnapshot의 섹션을 참조하세요. AWS CLI

### `restore-db-instance-from-s3`

다음 코드 예시에서는 `restore-db-instance-from-s3`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon S3의 백업에서 DB 인스턴스를 복원하려면

다음 `restore-db-instance-from-s3` 예제에서는 `my-backups` S3 버킷의 기존 백업`restored-test-instance`에서 라는 새 DB 인스턴스를 생성합니다.

```
aws rds restore-db-instance-from-s3 \
    --db-instance-identifier restored-test-instance \
    --allocated-storage 250 --db-instance-class db.m4.large --engine mysql \
    --master-username master --master-user-password secret99 \
    --s3-bucket-name my-backups --s3-ingestion-role-
arn arn:aws:iam::123456789012:role/my-role \
    --source-engine mysql --source-engine-version 5.6.27
```

- API 자세한 내용은 AWS CLI 명령 참조의 RestoreDbInstanceFromS3를 참조하세요.

### `restore-db-instance-to-point-in-time`

다음 코드 예시에서는 `restore-db-instance-to-point-in-time`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: DB 인스턴스를 특정 시점으로 복원하려면

다음 `restore-db-instance-to-point-in-time` 예제는 지정된 시간을 `restored-test-instance`기준으로 라는 `test-instance` 새 DB 인스턴스로 복원됩니다.

```
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
```

출력:

```
{
    "DBInstance": {
        "AllocatedStorage": 20,
        "DBInstanceArn": "arn:aws:rds:us-east-1:123456789012:db:restored-test-
instance",
        "DBInstanceStatus": "creating",
        "DBInstanceIdentifier": "restored-test-instance",
        ...some output omitted...
    }
}
```

자세한 내용은 Amazon RDS 사용 설명서의 DB 인스턴스를 지정된 시간으로 복원을 참조하세요.

예제 2: 복제된 백업에서 DB 인스턴스를 지정된 시간으로 복원하려면

다음 `restore-db-instance-to-point-in-time` 예제에서는 복제된 자동 백업에서 Oracle DB 인스턴스를 지정된 시간으로 복원합니다.

```
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-jkib2gfq5rv7replzadausbrktni2bn4example" \
    --target-db-instance-identifier myorclinstance-from-replicated-backup \
    --restore-time 2020-12-08T18:45:00.000Z
```

출력:

```
{
    "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": []
    }
}
```

자세한 내용은 Amazon RDS 사용 설명서의 복제된 백업에서 지정된 시간으로 복원을 참조하세요.

- 자세한 API 내용은 명령 참조RestoreDbInstanceToPointInTime의 섹션을 참조하세요. AWS CLI

### start-activity-stream

다음 코드 예시에서는 start-activity-stream을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터베이스 활동 스트림을 시작하려면

다음 start-activity-stream 예제에서는 라는 Aurora 클러스터를 모니터링하기 위해 비동기 활동 스트림을 시작합니다 my-pg-cluster.

```
aws rds start-activity-stream \
    --region us-east-1 \
    --mode async \
    --kms-key-id arn:aws:kms:us-east-1:1234567890123:key/a12c345d-6ef7-890g-
h123-456i789jk0l1 \
    --resource-arn arn:aws:rds:us-east-1:1234567890123:cluster:my-pg-cluster \
    --apply-immediately
```

출력:

```
{
    "KmsKeyId": "arn:aws:kms:us-east-1:1234567890123:key/a12c345d-6ef7-890g-
h123-456i789jk0l1",
    "KinesisStreamName": "aws-rds-das-cluster-0ABCDEFGHI1JKLM2NOPQ3R4S",
    "Status": "starting",
    "Mode": "async",
    "ApplyImmediately": true
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 데이터베이스 활동 스트림 시작을 참조하세요.

- 자세한 API 내용은 명령 참조StartActivityStream의 섹션을 참조하세요. AWS CLI

### start-db-cluster

다음 코드 예시에서는 start-db-cluster을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 클러스터를 시작하려면

다음 start-db-cluster 예제에서는 DB 클러스터와 해당 DB 인스턴스를 시작합니다.

```
aws rds start-db-cluster \
    --db-cluster-identifier mydbcluster
```

출력:

```
{
    "DBCluster": {
        "AllocatedStorage": 1,
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1e",
            "us-east-1b"
        ],
        "BackupRetentionPeriod": 1,
        "DatabaseName": "mydb",
        "DBClusterIdentifier": "mydbcluster",
        ...some output truncated...
    }
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 중지 및 시작을 참조하세요.

- 자세한 API 내용은 명령 참조StartDbCluster의 섹션을 참조하세요. AWS CLI

### start-db-instance-automated-backups-replication

다음 코드 예시에서는 start-db-instance-automated-backups-replication을 사용하는 방법을 보여 줍니다.

AWS CLI

리전 간 자동 백업을 활성화하려면

다음 start-db-instance-automated-backups-replication 예제는 미국 동부(버지니아 북부) 리전의 DB 인스턴스에서 미국 서부(오레곤)로 자동 백업을 복제합니다. 백업 보존 기간은 14일입니다.

```
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
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon RDS 사용 설명서의 리전 간 자동 백업 활성화를 참조하세요.

- 자세한 API 내용은 명령 참조StartDbInstanceAutomatedBackupsReplication의 섹션을 참조하세요. AWS CLI

### `start-db-instance`

다음 코드 예시에서는 `start-db-instance`을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 인스턴스를 시작하려면

다음 start-db-instance 예제에서는 지정된 DB 인스턴스를 시작합니다.

```
aws rds start-db-instance \
    --db-instance-identifier test-instance
```

출력:

```
{
    "DBInstance": {
        "DBInstanceStatus": "starting",
        ...some output truncated...
    }
}
```

- 자세한 API 내용은 명령 참조StartDbInstance의 섹션을 참조하세요. AWS CLI

### start-export-task

다음 코드 예시에서는 start-export-task을 사용하는 방법을 보여 줍니다.

AWS CLI

스냅샷을 Amazon S3로 내보내려면

다음 start-export-task 예제에서는 라는 DB 스냅샷을 라는 Amazon S3 버킷db5-snapshot-test으로 내보냅니다mybucket.

```
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
```

출력:

```
{
    "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
}
```

자세한 내용은 Amazon 사용 설명서의 Amazon S3 버킷으로 스냅샷 내보내기를 참조하세요. RDS

- 자세한 API 내용은 명령 참조StartExportTask의 섹션을 참조하세요. AWS CLI

### stop-activity-stream

다음 코드 예시에서는 stop-activity-stream을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터베이스 활동 스트림을 중지하려면

다음 stop-activity-stream 예제는 라는 Aurora 클러스터에서 활동 스트림을 중지합니다 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
```

출력:

```
{
    "KmsKeyId": "arn:aws:kms:us-east-1:1234567890123:key/a12c345d-6ef7-890g-
h123-456i789jk0l1",
    "KinesisStreamName": "aws-rds-das-cluster-0ABCDEFGHI1JKLM2NOPQ3R4S",
    "Status": "stopping"
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 활동 스트림 중지를 참조하세요.

- 자세한 API 내용은 명령 참조StopActivityStream의 섹션을 참조하세요. AWS CLI

### stop-db-cluster

다음 코드 예시에서는 stop-db-cluster을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 클러스터를 중지하려면

다음 stop-db-cluster 예제에서는 DB 클러스터와 해당 DB 인스턴스를 중지합니다.

```
aws rds stop-db-cluster \
    --db-cluster-identifier mydbcluster
```

출력:

```
{
    "DBCluster": {
        "AllocatedStorage": 1,
        "AvailabilityZones": [
            "us-east-1a",
            "us-east-1e",
            "us-east-1b"
        ],
        "BackupRetentionPeriod": 1,
        "DatabaseName": "mydb",
        "DBClusterIdentifier": "mydbcluster",
        ...some output truncated...
    }
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 Amazon Aurora DB 클러스터 중지 및 시작을 참조하세요.

- 자세한 API 내용은 명령 참조StopDbCluster의 섹션을 참조하세요. AWS CLI

### stop-db-instance-automated-backups-replication

다음 코드 예시에서는 stop-db-instance-automated-backups-replication을 사용하는 방법을 보여 줍니다.

AWS CLI

자동 백업 복제를 중지하려면

다음은 미국 서부(오레곤) 리전으로 자동 백업 복제를 stop-db-instance-automated-backups-replication 종료합니다. 복제된 백업은 설정된 백업 보존 기간에 따라 보존됩니다.

```
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"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon RDS 사용 설명서의 자동 백업 복제 중지를 참조하세요.

- 자세한 API 내용은 명령 참조StopDbInstanceAutomatedBackupsReplication의 섹션을 참조하세요. AWS CLI

### stop-db-instance

다음 코드 예시에서는 stop-db-instance을 사용하는 방법을 보여 줍니다.

AWS CLI

DB 인스턴스를 중지하려면

다음 stop-db-instance 예제에서는 지정된 DB 인스턴스를 중지합니다.

```
aws rds stop-db-instance \
    --db-instance-identifier test-instance
```

출력:

```
{
    "DBInstance": {
        "DBInstanceStatus": "stopping",
        ...some output truncated...
    }
}
```

- 자세한 API 내용은 명령 참조StopDbInstance의 섹션을 참조하세요. AWS CLI

### switchover-blue-green-deployment

다음 코드 예시에서는 switchover-blue-green-deployment을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: RDS DB 인스턴스에 대한 블루/그린 배포 전환

다음 switchover-blue-green-deployment 예제에서는 지정된 그린 환경을 새 프로덕션 환경으로 승격합니다.

```
aws rds switchover-blue-green-deployment \
    --blue-green-deployment-identifier bgd-wi89nwzglccsfake \
    --switchover-timeout 300
```

출력:

```
{
    "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-
blhi1e",
        "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-blhi1e",
                "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"
    }
}
```

자세한 내용은 Amazon RDS 사용 설명서의 블루/그린 배포 전환을 참조하세요.

예제 2: Aurora MySQL DB 클러스터에 대한 블루/그린 배포를 승격하려면

다음 `switchover-blue-green-deployment` 예제에서는 지정된 그린 환경을 새 프로덕션 환경으로 승격합니다.

```
aws rds switchover-blue-green-deployment \
    --blue-green-deployment-identifier bgd-wi89nwzglccsfake \
    --switchover-timeout 300
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Aurora 사용 설명서의 블루/그린 배포 전환을 참조하세요.

- 자세한 API 내용은 명령 참조SwitchoverBlueGreenDeployment의 섹션을 참조하세요. AWS CLI

## 를 사용한 Amazon RDS Data Service 예제 AWS CLI

다음 코드 예제에서는 Amazon RDS Data Service와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### batch-execute-statement

다음 코드 예시에서는 batch-execute-statement을 사용하는 방법을 보여 줍니다.

AWS CLI

배치 SQL 문을 실행하려면

다음 batch-execute-statement 예제에서는 파라미터 세트가 있는 데이터 배열을 통해 배치 SQL 문을 실행합니다.

```
aws rds-data batch-execute-statement \
    --resource-arn "arn:aws:rds:us-west-2:123456789012:cluster:mydbcluster" \
    --database "mydb" \
    --secret-arn "arn:aws:secretsmanager:us-west-2:123456789012:secret:mysecret" \
    --sql "insert into mytable values (:id, :val)" \
    --parameter-sets "[[{\"name\": \"id\", \"value\": {\"longValue\": 1}},{\"name\":
 \"val\", \"value\": {\"stringValue\": \"ValueOne\"}}],
       [{\"name\": \"id\", \"value\": {\"longValue\": 2}},{\"name\": \"val\",
 \"value\": {\"stringValue\": \"ValueTwo\"}}],
       [{\"name\": \"id\", \"value\": {\"longValue\": 3}},{\"name\": \"val\",
 \"value\": {\"stringValue\": \"ValueThree\"}}]]"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon RDS 사용 설명서의 Aurora ServerlessAPI용 데이터 사용을 참조하세요.

- 자세한 API 내용은 명령 참조BatchExecuteStatement의 섹션을 참조하세요. AWS CLI

### begin-transaction

다음 코드 예시에서는 begin-transaction을 사용하는 방법을 보여 줍니다.

AWS CLI

SQL 트랜잭션을 시작하려면

다음 begin-transaction 예제에서는 SQL 트랜잭션을 시작합니다.

```
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"
```

출력:

```
{
    "transactionId": "ABC1234567890xyz"
}
```

자세한 내용은 Amazon RDS 사용 설명서의 Aurora ServerlessAPI용 데이터 사용을 참조하세요.

- 자세한 API 내용은 명령 참조BeginTransaction의 섹션을 참조하세요. AWS CLI

### commit-transaction

다음 코드 예시에서는 commit-transaction을 사용하는 방법을 보여 줍니다.

AWS CLI

SQL 트랜잭션을 커밋하려면

다음 commit-transaction 예제에서는 지정된 SQL 트랜잭션을 종료하고 그 일부로 수행한 변경 사항을 커밋합니다.

```
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"
```

출력:

```
{
    "transactionStatus": "Transaction Committed"
}
```

자세한 내용은 Amazon RDS 사용 설명서의 Aurora ServerlessAPI용 데이터 사용을 참조하세요.

- 자세한 API 내용은 명령 참조CommitTransaction의 섹션을 참조하세요. AWS CLI

### execute-statement

다음 코드 예시에서는 execute-statement을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 트랜잭션의 일부인 SQL 문을 실행하려면

다음 execute-statement 예제에서는 트랜잭션의 일부인 SQL 문을 실행합니다.

```
aws rds-data execute-statement \
    --resource-arn "arn:aws:rds:us-west-2:123456789012:cluster:mydbcluster" \
    --database "mydb" \
    --secret-arn "arn:aws:secretsmanager:us-west-2:123456789012:secret:mysecret" \
    --sql "update mytable set quantity=5 where id=201" \
    --transaction-id "ABC1234567890xyz"
```

출력:

```
{
    "numberOfRecordsUpdated": 1
}
```

예제 2: 파라미터로 SQL 문 실행

다음 execute-statement 예제에서는 파라미터가 있는 SQL 문을 실행합니다.

```
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\"}}]"
```

출력:

```
{
    "numberOfRecordsUpdated": 1
}
```

자세한 내용은 Amazon RDS 사용 설명서의 Aurora ServerlessAPI용 데이터 사용을 참조하세요.

- 자세한 API 내용은 명령 참조ExecuteStatement의 섹션을 참조하세요. AWS CLI

**rollback-transaction**

다음 코드 예시에서는 rollback-transaction을 사용하는 방법을 보여 줍니다.

AWS CLI

SQL 트랜잭션을 롤백하려면

다음 rollback-transaction 예제에서는 지정된 SQL 트랜잭션을 롤백합니다.

```
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"
```

출력:

```
{
    "transactionStatus": "Rollback Complete"
}
```

자세한 내용은 Amazon RDS 사용 설명서의 Aurora ServerlessAPI용 데이터 사용을 참조하세요.

- 자세한 API 내용은 명령 참조RollbackTransaction의 섹션을 참조하세요. AWS CLI

## 를 사용한 Amazon RDS 성능 개선 도우미 예제 AWS CLI

다음 코드 예제에서는 Amazon RDS Performance Insights와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### describe-dimension-keys

다음 코드 예시에서는 describe-dimension-keys을 사용하는 방법을 보여 줍니다.

AWS CLI

차원 키를 설명하려면

이 예제에서는 모든 대기 이벤트의 이름을 요청합니다. 데이터는 이벤트 이름 및 지정된 기간 동안의 해당 이벤트의 집계 값으로 요약됩니다.

명령:

```
aws pi describe-dimension-keys --service-type RDS --identifier db-
LKCGOBK26374TPTDFXOIWVCPPM --start-time 1527026400 --end-time 1527080400 --
metric db.load.avg --group-by '{"Group":"db.wait_event"}'
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeDimensionKeys의 섹션을 참조하세요. AWS CLI

### `get-resource-metrics`

다음 코드 예시에서는 `get-resource-metrics`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 지표를 가져오려면

이 예제에서는 db.wait_event 차원 그룹과 해당 그룹 내의 db.wait_event.name 차원에 대한 데이터 포인트를 요청합니다. 응답에서 관련 데이터 포인트는 요청된 차원(db.wait_event.name )별로 그룹화됩니다.

명령:

```
aws pi get-resource-metrics --service-type RDS --identifier db-
LKCGOBK26374TPTDFXOIWVCPPM --start-time 1527026400 --end-time 1527080400 --period-
in-seconds 300 --metric db.load.avg --metric-queries file://metric-queries.json
```

에 대한 인수는 JSON 파일 에 `--metric-queries` 저장됩니다`metric-queries.json`. 해당 파일의 내용은 다음과 같습니다.

```
[
    {
        "Metric": "db.load.avg",
        "GroupBy": {
            "Group":"db.wait_event"
        }
    }
]
```

출력:

```
{
    "AlignedEndTime": 1.5270804E9,
    "AlignedStartTime": 1.5270264E9,
    "Identifier": "db-LKCGOBK26374TPTDFXOIWVCPPM",
    "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...>
    ]
}
```

- 자세한 API 내용은 명령 참조GetResourceMetrics의 섹션을 참조하세요. AWS CLI

## 를 사용한 Amazon Redshift 예제 AWS CLI

다음 코드 예제에서는 Amazon Redshift와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### `accept-reserved-node-exchange`

다음 코드 예시에서는 `accept-reserved-node-exchange`을 사용하는 방법을 보여 줍니다.

AWS CLI

예약 노드 교환을 수락하려면

다음 `accept-reserved-node-exchange` 예제에서는 DC1 예약 노드를 DC2 예약 노드로 교환하는 것을 허용합니다.

```
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
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Redshift 클러스터 관리 안내서의 를 사용하여 예약 노드 업그레이드 AWS CLI 섹션을 참조하세요.

- 자세한 API 내용은 명령 참조AcceptReservedNodeExchange의 섹션을 참조하세요. AWS CLI

### `authorize-cluster-security-group-ingress`

다음 코드 예시에서는 `authorize-cluster-security-group-ingress`을 사용하는 방법을 보여 줍니다.

AWS CLI

EC2 보안에 대한 액세스 권한 부여 GroupThis 예제는 명명된 Amazon EC2 보안 그룹에 대한 액세스 권한을 부여합니다.명령:

```
aws redshift authorize-cluster-security-group-ingress --cluster-security-group-name
 mysecuritygroup --ec2-security-group-name myec2securitygroup --ec2-security-group-
owner-id 123445677890
```

CIDR rangeThis 예제에 대한 액세스 권한을 부여하면 CIDR 범위에 대한 액세스 권한을 부여합니다.명령:

```
aws redshift authorize-cluster-security-group-ingress --cluster-security-group-name
 mysecuritygroup --cidrip 192.168.100.100/32
```

- 자세한 API 내용은 명령 참조AuthorizeClusterSecurityGroupIngress의 섹션을 참조하세요. AWS CLI

### `authorize-snapshot-access`

다음 코드 예시에서는 `authorize-snapshot-access`을 사용하는 방법을 보여 줍니다.

AWS CLI

SnapshotThis 예제를 복원할 AWS 수 있는 권한 부여는 스냅샷을 복원444455556666할 수 있는 권한을 AWS 계정에 부여합니다`my-snapshot-id`. 기본적으로 출력은 JSON 형식입니다.명령:

```
aws redshift authorize-snapshot-access --snapshot-id my-snapshot-id --account-with-
restore-access 444455556666
```

결과:

```
{
   "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"
   }
}
```

- 자세한 API 내용은 명령 참조AuthorizeSnapshotAccess의 섹션을 참조하세요. AWS CLI

### batch-delete-cluster-snapshots

다음 코드 예시에서는 batch-delete-cluster-snapshots을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터 스냅샷 세트를 삭제하려면

다음 batch-delete-cluster-snapshots 예제에서는 수동 클러스터 스냅샷 세트를 삭제합니다.

```
aws redshift batch-delete-cluster-snapshots \
        --
identifiers SnapshotIdentifier=mycluster-2019-11-06-14-12 SnapshotIdentifier=mycluster-2019-
```

출력:

```
{
    "Resources": [
        "mycluster-2019-11-06-14-12",
        "mycluster-2019-11-06-14-20"
    ]
}
```

자세한 내용은 Amazon Redshift 클러스터 관리 안내서의 Amazon Redshift 스냅샷을 참조하세요.

- 자세한 API 내용은 명령 참조BatchDeleteClusterSnapshots의 섹션을 참조하세요. AWS CLI

### batch-modify-cluster-snapshots

다음 코드 예시에서는 batch-modify-cluster-snapshots을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터 스냅샷 세트를 수정하려면

다음 batch-modify-cluster-snapshots 예제에서는 클러스터 스냅샷 세트에 대한 설정을 수정합니다.

```
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
```

출력:

```
{
    "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
        }
}
```

자세한 내용은 Amazon Redshift 클러스터 관리 안내서의 Amazon Redshift 스냅샷을 참조하세요.

- 자세한 API 내용은 명령 참조BatchModifyClusterSnapshots의 섹션을 참조하세요. AWS CLI

### `cancel-resize`

다음 코드 예시에서는 `cancel-resize`을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터 크기 조정을 취소하려면

다음 `cancel-resize` 예제에서는 클러스터에 대한 클래식 크기 조정 작업을 취소합니다.

```
aws redshift cancel-resize \
    --cluster-identifier mycluster
```

출력:

```
{
    "TargetNodeType": "dc2.large",
    "TargetNumberOfNodes": 2,
    "TargetClusterType": "multi-node",
    "Status": "CANCELLING",
    "ResizeType": "ClassicResize",
    "TargetEncryptionType": "NONE"
}
```

자세한 내용은 Amazon Redshift 클러스터 관리 안내서의 Amazon Redshift에서 클러스터 크기 조정을 참조하세요.

- 자세한 API 내용은 명령 참조CancelResize의 섹션을 참조하세요. AWS CLI

### copy-cluster-snapshot

다음 코드 예시에서는 copy-cluster-snapshot을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 클러스터 설명 가져오기 VersionsThis 예제는 모든 클러스터 버전에 대한 설명을 반환합니다. 기본적으로 출력은 JSON 형식입니다.명령:

```
aws redshift copy-cluster-snapshot --source-snapshot-identifier
 cm:examplecluster-2013-01-22-19-27-58 --target-snapshot-identifier my-saved-
snapshot-copy
```

결과:

```
{
   "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"
   }
}
```

- 자세한 API 내용은 명령 참조CopyClusterSnapshot의 섹션을 참조하세요. AWS CLI

### create-cluster-parameter-group

다음 코드 예시에서는 create-cluster-parameter-group을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터 파라미터 생성 GroupThis 예제는 새 클러스터 파라미터 그룹을 생성합니다.명령:

```
aws redshift create-cluster-parameter-group --parameter-group-name
 myclusterparametergroup --parameter-group-family redshift-1.0 --description "My
 first cluster parameter group"
```

결과:

```
{
   "ClusterParameterGroup": {
      "ParameterGroupFamily": "redshift-1.0",
      "Description": "My first cluster parameter group",
      "ParameterGroupName": "myclusterparametergroup"
   },
   "ResponseMetadata": {
      "RequestId": "739448f0-64cc-11e2-8f7d-3b939af52818"
   }
}
```

- 자세한 API 내용은 명령 참조CreateClusterParameterGroup의 섹션을 참조하세요. AWS CLI

### `create-cluster-security-group`

다음 코드 예시에서는 `create-cluster-security-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터 보안 GroupThis 예제를 생성하면 새 클러스터 보안 그룹이 생성됩니다. 기본적으로 출력은 JSON 형식입니다.명령:

```
aws redshift create-cluster-security-group --cluster-security-group-name
 mysecuritygroup --description "This is my cluster security group"
```

결과:

```
{
   "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"
      }
   }
}
```

`--output text` 옵션을 사용하여 텍스트 형식으로 동일한 정보를 얻을 수도 있습니다. 명령:

`--output text` 옵션. 명령:

옵션. 명령:

```
aws redshift create-cluster-security-group --cluster-security-group-name
 mysecuritygroup --description "This is my cluster security group" --output text
```

결과:

```
This is my cluster security group   300454760768    mysecuritygroup
a0c0bfab-343a-11e2-95d2-c3dc9fe8ab57
```

- 자세한 API 내용은 명령 참조CreateClusterSecurityGroup의 섹션을 참조하세요. AWS CLI

### `create-cluster-snapshot`

다음 코드 예시에서는 `create-cluster-snapshot`을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터 생성 SnapshotThis 예제는 새 클러스터 스냅샷을 생성합니다. 기본적으로 출력은 JSON 형식입니다.명령:

```
aws redshift create-cluster-snapshot --cluster-identifier mycluster --snapshot-
identifier my-snapshot-id
```

결과:

```
{
   "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"
   }
}
```

- 자세한 API 내용은 명령 참조CreateClusterSnapshot의 섹션을 참조하세요. AWS CLI

### `create-cluster-subnet-group`

다음 코드 예시에서는 `create-cluster-subnet-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터 서브넷 생성 GroupThis 예제는 새 클러스터 서브넷 그룹을 생성합니다.명령:

```
aws redshift create-cluster-subnet-group --cluster-subnet-group-name mysubnetgroup
 --description "My subnet group" --subnet-ids subnet-763fdd1c
```

결과:

```
{
   "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"
   }
}
```

- 자세한 API 내용은 명령 참조CreateClusterSubnetGroup의 섹션을 참조하세요. AWS CLI

### `create-cluster`

다음 코드 예시에서는 `create-cluster`을 사용하는 방법을 보여 줍니다.

AWS CLI

최소 수준의 클러스터 생성 ParametersThis 예제는 최소 파라미터 집합으로 클러스터를 생성합니다. 기본적으로 출력은 JSON 형식입니다.명령:

```
aws redshift create-cluster --node-type dw.hs1.xlarge --number-of-nodes 2 --master-
username adminuser --master-user-password TopSecret1 --cluster-identifier mycluster
```

결과:

```
{
   "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"
      }
}
```

- 자세한 API 내용은 명령 참조CreateCluster의 섹션을 참조하세요. AWS CLI

### `create-event-subscription`

다음 코드 예시에서는 `create-event-subscription`을 사용하는 방법을 보여 줍니다.

AWS CLI

이벤트에 대한 알림 구독을 생성하려면

다음 `create-event-subscription` 예제에서는 이벤트 알림 구독을 생성합니다.

```
aws redshift create-event-subscription \
    --subscription-name mysubscription \
    --sns-topic-arn arn:aws:sns:us-west-2:123456789012:MySNStopic \
    --source-type cluster \
    --source-ids mycluster
```

출력:

```
{
```

```
        "EventSubscription": {
        "CustomerAwsId": "123456789012",
        "CustSubscriptionId": "mysubscription",
        "SnsTopicArn": "arn:aws:sns:us-west-2:123456789012:MySNStopic",
        "Status": "active",
        "SubscriptionCreationTime": "2019-12-09T20:05:19.365Z",
        "SourceType": "cluster",
        "SourceIdsList": [
            "mycluster"
        ],
        "EventCategoriesList": [],
        "Severity": "INFO",
        "Enabled": true,
        "Tags": []
    }
}
```

자세한 내용은 Amazon Redshift 클러스터 관리 안내서의 Amazon Redshift 이벤트 알림 구독을 참조하세요.

- 자세한 API 내용은 명령 참조CreateEventSubscription의 섹션을 참조하세요. AWS CLI

### `create-hsm-client-certificate`

다음 코드 예시에서는 `create-hsm-client-certificate`을 사용하는 방법을 보여 줍니다.

AWS CLI

HSM 클라이언트 인증서를 생성하려면

다음 `create-hsm-client-certificate` 예제에서는 클러스터가 에 연결하는 데 사용할 수 있는 HSM 클라이언트 인증서를 생성합니다HSM.

```
aws redshift create-hsm-client-certificate \
    --hsm-client-certificate-identifier myhsmclientcert
```

출력:

```
{
    "HsmClientCertificate": {
        "HsmClientCertificateIdentifier": "myhsmclientcert",
        "HsmClientCertificatePublicKey": "-----BEGIN CERTIFICATE-----
```

```
        MIICiEXAMPLECQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
        VVMxCzAJBgNVBAgTEXAMPLEwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
        b24xFDASBgNVBAsTC0lBTSBDb25EXAMPLEIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
        BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb2EXAMPLETEwNDI1MjA0NTIxWhcN
        MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBEXAMPLEMRAwDgYD
        EXAMPLETZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
        b2xlMRIwEAEXAMPLEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFt
        YXpvbi5jb20wgZ8wDQYJKEXAMPLEAQEBBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ
        21uUSfwfEvySWtC2XADZ4nB+BLYgVIk6EXAMPLE3G93vUEIO3IyNoH/f0wYK8m9T
        rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugEXAMPLEzZswY6786m86gpE
        Ibb3OhjZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEEXAMPLEEAtCu4
        nUhVVxYUEXAMPLEh8Mg9q6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
        FFBjvSfpJIlJ00zbhNYS5f6GEXAMPLEl0ZxBHjJnyp378OD8uTs7fLvjx79LjSTb
        NYiytVbZPQUQ5Yaxu2jXnimvw3rEXAMPLE=-----END CERTIFICATE-----\n",
    "Tags": []
    }
}
```

자세한 내용은 Amazon Redshift 클러스터 관리 안내서의 Amazon Redshift API 권한 참조를 참조하세요.

- 자세한 API 내용은 명령 참조CreateHsmClientCertificate의 섹션을 참조하세요. AWS CLI

### `create-hsm-configuration`

다음 코드 예시에서는 `create-hsm-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

HSM 구성을 생성하려면

다음 `create-hsm-configuration` 예제에서는 클러스터가 하드웨어 보안 모듈()에 데이터베이스 암호화 키를 저장하고 사용하는 데 필요한 정보를 포함하는 지정된 HSM 구성을 생성합니다 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
```

출력:

```
{
    "HsmConfiguration": {
        "HsmConfigurationIdentifier": "myhsmconnection",
        "Description": "My HSM connection",
        "HsmIpAddress": "192.0.2.09",
        "HsmPartitionName": "myhsmpartition",
        "Tags": []
    }
}
```

- 자세한 API 내용은 명령 참조CreateHsmConfiguration의 섹션을 참조하세요. AWS CLI

### `create-snapshot-copy-grant`

다음 코드 예시에서는 `create-snapshot-copy-grant`을 사용하는 방법을 보여 줍니다.

AWS CLI

스냅샷 복사 권한 부여를 생성하려면

다음 `create-snapshot-copy-grant` 예제에서는 스냅샷 복사 권한 부여를 생성하고 대상 AWS 리전에서 복사된 스냅샷을 암호화합니다.

```
aws redshift create-snapshot-copy-grant \
    --snapshot-copy-grant-name mysnapshotcopygrantname
```

출력:

```
{
    "SnapshotCopyGrant": {
        "SnapshotCopyGrantName": "mysnapshotcopygrantname",
        "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/
bPxRfih3yCo8nvbEXAMPLEKEY",
        "Tags": []
    }
}
```

자세한 내용은 Amazon Redshift 클러스터 관리 안내서의 Amazon Redshift 데이터베이스 암호화를 참조하세요.

- 자세한 API 내용은 명령 참조CreateSnapshotCopyGrant의 섹션을 참조하세요. AWS CLI

### create-snapshot-schedule

다음 코드 예시에서는 create-snapshot-schedule을 사용하는 방법을 보여 줍니다.

AWS CLI

스냅샷 일정을 생성하려면

다음 create-snapshot-schedule 예제에서는 지정된 설명과 12시간마다의 속도로 스냅샷 일정을 생성합니다.

```
aws redshift create-snapshot-schedule \
    --schedule-definitions "rate(12 hours)" \
    --schedule-identifier mysnapshotschedule \
    --schedule-description "My schedule description"
```

출력:

```
{
    "ScheduleDefinitions": [
        "rate(12 hours)"
    ],
    "ScheduleIdentifier": "mysnapshotschedule",
    "ScheduleDescription": "My schedule description",
    "Tags": []
}
```

자세한 내용은 Amazon Redshift 클러스터 관리 안내서의 자동 스냅샷 일정을 참조하세요.

- 자세한 API 내용은 명령 참조CreateSnapshotSchedule의 섹션을 참조하세요. AWS CLI

### create-tags

다음 코드 예시에서는 create-tags을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터에 대한 태그를 생성하려면

다음 create-tags 예제에서는 지정된 태그 키/값 페어를 지정된 클러스터에 추가합니다.

```
aws redshift create-tags \
    --resource-name arn:aws:redshift:us-west-2:123456789012:cluster:mycluster \
    --tags "Key"="mytags","Value"="tag1"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Redshift 클러스터 관리 안내서의 Amazon Redshift에서 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조CreateTags의 섹션을 참조하세요. AWS CLI

### delete-cluster-parameter-group

다음 코드 예시에서는 delete-cluster-parameter-group을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터 파라미터 삭제 GroupThis 예제는 클러스터 파라미터 그룹을 삭제합니다.명령:

```
aws redshift delete-cluster-parameter-group --parameter-group-name
 myclusterparametergroup
```

- 자세한 API 내용은 명령 참조DeleteClusterParameterGroup의 섹션을 참조하세요. AWS CLI

### delete-cluster-security-group

다음 코드 예시에서는 delete-cluster-security-group을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터 보안 삭제 GroupThis 예제는 클러스터 보안 그룹을 삭제합니다.명령:

```
aws redshift delete-cluster-security-group --cluster-security-group-name
 mysecuritygroup
```

- 자세한 API 내용은 명령 참조DeleteClusterSecurityGroup의 섹션을 참조하세요. AWS CLI

### delete-cluster-snapshot

다음 코드 예시에서는 delete-cluster-snapshot을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터 삭제 SnapshotThis 예제는 클러스터 스냅샷을 삭제합니다.명령:

```
aws redshift delete-cluster-snapshot --snapshot-identifier my-snapshot-id
```

- 자세한 API 내용은 명령 참조DeleteClusterSnapshot의 섹션을 참조하세요. AWS CLI

### `delete-cluster-subnet-group`

다음 코드 예시에서는 `delete-cluster-subnet-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터 서브넷 삭제 GroupThis 예제는 클러스터 서브넷 그룹을 삭제합니다.명령:

```
aws redshift delete-cluster-subnet-group --cluster-subnet-group-name mysubnetgroup
```

결과:

```
{
   "ResponseMetadata": {
      "RequestId": "253fbffd-6993-11e2-bc3a-47431073908a"
   }
}
```

- 자세한 API 내용은 명령 참조DeleteClusterSubnetGroup의 섹션을 참조하세요. AWS CLI

### `delete-cluster`

다음 코드 예시에서는 `delete-cluster`을 사용하는 방법을 보여 줍니다.

AWS CLI

최종 클러스터가 없는 클러스터 삭제 SnapshotThis 예제는 클러스터를 삭제하여 최종 클러스터 스냅샷이 생성되지 않도록 데이터 삭제를 강제합니다.명령:

```
aws redshift delete-cluster --cluster-identifier mycluster --skip-final-cluster-
snapshot
```

클러스터 삭제, 최종 클러스터 허용 SnapshotThis 예제는 클러스터를 삭제하지만 최종 클러스터 스냅샷을 지정합니다.명령:

```
aws redshift delete-cluster --cluster-identifier mycluster --final-cluster-snapshot-identifier myfinalsnapshot
```

- 자세한 API 내용은 명령 참조DeleteCluster의 섹션을 참조하세요. AWS CLI

### delete-event-subscription

다음 코드 예시에서는 delete-event-subscription을 사용하는 방법을 보여 줍니다.

AWS CLI

이벤트 구독을 삭제하려면

다음 delete-event-subscription 예제에서는 지정된 이벤트 알림 구독을 삭제합니다.

```
aws redshift delete-event-subscription \
    --subscription-name mysubscription
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Redshift 클러스터 관리 안내서의 Amazon Redshift 이벤트 알림 구독을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteEventSubscription의 섹션을 참조하세요. AWS CLI

### delete-hsm-client-certificate

다음 코드 예시에서는 delete-hsm-client-certificate을 사용하는 방법을 보여 줍니다.

AWS CLI

HSM 클라이언트 인증서를 삭제하려면

다음 delete-hsm-client-certificate 예제에서는 HSM 클라이언트 인증서를 삭제합니다.

```
aws redshift delete-hsm-client-certificate \
    --hsm-client-certificate-identifier myhsmclientcert
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Redshift 클러스터 관리 안내서의 Amazon Redshift API 권한 참조를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteHsmClientCertificate의 섹션을 참조하세요. AWS CLI

### delete-hsm-configuration

다음 코드 예시에서는 delete-hsm-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

HSM 구성을 삭제하려면

다음 delete-hsm-configuration 예제에서는 현재 AWS 계정에서 지정된 HSM 구성을 삭제합니다.

```
aws redshift delete-hsm-configuration /
    --hsm-configuration-identifier myhsmconnection
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeleteHsmConfiguration의 섹션을 참조하세요. AWS CLI

### delete-scheduled-action

다음 코드 예시에서는 delete-scheduled-action을 사용하는 방법을 보여 줍니다.

AWS CLI

예약된 작업을 삭제하려면

다음 delete-scheduled-action 예제에서는 지정된 예약된 작업을 삭제합니다.

```
aws redshift delete-scheduled-action \
    --scheduled-action-name myscheduledaction
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeleteScheduledAction의 섹션을 참조하세요. AWS CLI

### delete-snapshot-copy-grant

다음 코드 예시에서는 delete-snapshot-copy-grant을 사용하는 방법을 보여 줍니다.

AWS CLI

스냅샷 복사 권한 부여를 삭제하려면

다음 delete-snapshot-copy-grant 예제에서는 지정된 스냅샷 복사 권한 부여를 삭제합니다.

```
aws redshift delete-snapshot-copy-grant \
    --snapshot-copy-grant-name mysnapshotcopygrantname
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Redshift 클러스터 관리 안내서의 Amazon Redshift 데이터베이스 암호화를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteSnapshotCopyGrant의 섹션을 참조하세요. AWS CLI

### delete-snapshot-schedule

다음 코드 예시에서는 delete-snapshot-schedule을 사용하는 방법을 보여 줍니다.

AWS CLI

스냅샷 일정을 삭제하려면

다음 delete-snapshot-schedule 예제에서는 지정된 스냅샷 일정을 삭제합니다. 일정을 삭제하기 전에 클러스터 연결을 해제해야 합니다.

```
aws redshift delete-snapshot-schedule \
    --schedule-identifier mysnapshotschedule
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Redshift 클러스터 관리 안내서의 자동 스냅샷 일정을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteSnapshotSchedule의 섹션을 참조하세요. AWS CLI

### delete-tags

다음 코드 예시에서는 delete-tags을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터에서 태그를 삭제하려면

다음 delete-tags 예제에서는 지정된 클러스터에서 지정된 키 이름을 가진 태그를 삭제합니다.

```
aws redshift delete-tags \
    --resource-name arn:aws:redshift:us-west-2:123456789012:cluster:mycluster \
    --tag-keys "clustertagkey" "clustertagvalue"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Redshift 클러스터 관리 안내서의 Amazon Redshift에서 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteTags의 섹션을 참조하세요. AWS CLI

### describe-account-attributes

다음 코드 예시에서는 describe-account-attributes을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 계정의 속성을 설명하려면

다음 describe-account-attributes 예제에서는 호출 AWS 계정에 연결된 속성을 표시합니다.

```
aws redshift describe-account-attributes
```

출력:

```
{
    "AccountAttributes": [
        {
            "AttributeName": "max-defer-maintenance-duration",
            "AttributeValues": [
                {
                    "AttributeValue": "45"
                }
            ]
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeAccountAttributes의 섹션을 참조하세요. AWS CLI

### `describe-cluster-db-revisions`

다음 코드 예시에서는 `describe-cluster-db-revisions`을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터의 DB 개정을 설명하려면

다음 `describe-cluster-db-revisions` 예제에서는 지정된 클러스터에 대한 `ClusterDbRevision` 객체 배열의 세부 정보를 표시합니다.

```
aws redshift describe-cluster-db-revisions \
    --cluster-identifier mycluster
```

출력:

```
{
    "ClusterDbRevisions": [
        {
            "ClusterIdentifier": "mycluster",
            "CurrentDatabaseRevision": "11420",
            "DatabaseRevisionReleaseDate": "2019-11-22T16:43:49.597Z",
            "RevisionTargets": []
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeClusterDbRevisions의 섹션을 참조하세요. AWS CLI

### `describe-cluster-parameter-groups`

다음 코드 예시에서는 `describe-cluster-parameter-groups`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 클러스터 파라미터 설명 가져오기 GroupsThis 예제는 열 헤더와 함께 계정에 대한 모든 클러스터 파라미터 그룹에 대한 설명을 반환합니다. 기본적으로 출력은 JSON 형식입니다.명령:

```
aws redshift describe-cluster-parameter-groups
```

결과:

```
{
   "ParameterGroups": [
      {
         "ParameterGroupFamily": "redshift-1.0",
         "Description": "My first cluster parameter group",
         "ParameterGroupName": "myclusterparametergroup"
      } ],
   "ResponseMetadata": {
      "RequestId": "8ceb8f6f-64cc-11e2-bea9-49e0ce183f07"
   }
}
```

`--output text` 옵션을 사용하여 텍스트 형식으로 동일한 정보를 얻을 수도 있습니다. 명령:

`--output text` 옵션. 명령:

옵션. 명령:

```
aws redshift describe-cluster-parameter-groups --output text
```

결과:

```
redshift-1.0        My first cluster parameter group        myclusterparametergroup
RESPONSEMETADATA    9e665a36-64cc-11e2-8f7d-3b939af52818
```

- 자세한 API 내용은 명령 참조DescribeClusterParameterGroups의 섹션을 참조하세요. AWS CLI

### `describe-cluster-parameters`

다음 코드 예시에서는 `describe-cluster-parameters`을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 클러스터 파라미터의 파라미터 검색 GroupThis 예제는 명명된 파라미터 그룹의 파라미터를 검색합니다. 기본적으로 출력은 JSON 형식입니다.명령:

```
aws redshift describe-cluster-parameters --parameter-group-name
 myclusterparametergroup
```

결과:

```
{
   "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...)
   ]
}
```

`--output text` 옵션을 사용하여 텍스트 형식으로 동일한 정보를 얻을 수도 있습니다. 명령:

`--output text` 옵션. 명령:

옵션. 명령:

```
aws redshift describe-cluster-parameters --parameter-group-name
 myclusterparametergroup --output text
```

결과:

```
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
```

- 자세한 API 내용은 명령 참조DescribeClusterParameters의 섹션을 참조하세요. AWS CLI

### `describe-cluster-security-groups`

다음 코드 예시에서는 `describe-cluster-security-groups`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 클러스터 보안 설명 가져오기 GroupsThis 예제는 계정에 대한 모든 클러스터 보안 그룹에 대한 설명을 반환합니다. 기본적으로 출력은 JSON 형식입니다.명령:

```
aws redshift describe-cluster-security-groups
```

결과:

```
{
   "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...)
      ]
}
```

- 자세한 API 내용은 명령 참조DescribeClusterSecurityGroups의 섹션을 참조하세요. AWS CLI

### `describe-cluster-snapshots`

다음 코드 예시에서는 `describe-cluster-snapshots`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 클러스터에 대한 설명 가져오기 SnapshotsThis 예제는 계정에 대한 모든 클러스터 스냅샷에 대한 설명을 반환합니다. 기본적으로 출력은 JSON 형식입니다.명령:

```
aws redshift describe-cluster-snapshots
```

결과:

```
{
   "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...)
```

- 자세한 API 내용은 명령 참조DescribeClusterSnapshots의 섹션을 참조하세요. AWS CLI

### describe-cluster-subnet-groups

다음 코드 예시에서는 describe-cluster-subnet-groups을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 클러스터 서브넷에 대한 설명 가져오기 GroupsThis 예제는 모든 클러스터 서브넷 그룹에 대한 설명을 반환합니다. 기본적으로 출력은 JSON 형식입니다.명령:

```
aws redshift describe-cluster-subnet-groups
```

결과:

```
{
   "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"
   }
}
```

- 자세한 API 내용은 명령 참조DescribeClusterSubnetGroups의 섹션을 참조하세요. AWS CLI

### `describe-cluster-tracks`

다음 코드 예시에서는 `describe-cluster-tracks`을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터 트랙을 설명하려면

다음 describe-cluster-tracks 예제에서는 사용 가능한 유지 관리 트랙의 세부 정보를 보여 줍니다.

```
aws redshift describe-cluster-tracks \
    --maintenance-track-name current
```

출력:

```
{
    "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"
                        }
                    ]
                }
            ]
        }
    ]
}
```

자세한 내용은 Amazon Redshift 클러스터 관리 안내서의 클러스터 유지 관리 트랙 선택을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeClusterTracks의 섹션을 참조하세요. AWS CLI

### `describe-cluster-versions`

다음 코드 예시에서는 `describe-cluster-versions`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 클러스터 설명 가져오기 VersionsThis 예제는 모든 클러스터 버전에 대한 설명을 반환합니다. 기본적으로 출력은 JSON 형식입니다.명령:

```
aws redshift describe-cluster-versions
```

결과:

```
{
   "ClusterVersions": [
      {
      "ClusterVersion": "1.0",
      "Description": "Initial release",
      "ClusterParameterGroupFamily": "redshift-1.0"
      } ],
   "ResponseMetadata": {
      "RequestId": "16a53de3-64cc-11e2-bec0-17624ad140dd"
   }
}
```

- 자세한 API 내용은 명령 참조DescribeClusterVersions의 섹션을 참조하세요. AWS CLI

### `describe-clusters`

다음 코드 예시에서는 `describe-clusters`을 사용하는 방법을 보여 줍니다.

AWS CLI

Get a Description of All ClustersThis example는 계정에 대한 모든 클러스터에 대한 설명을 반환합니다. 기본적으로 출력은 JSON 형식입니다.명령:

```
aws redshift describe-clusters
```

결과:

```
{
   "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"
   }
}
```

`--output text` 옵션을 사용하여 텍스트 형식으로 동일한 정보를 얻을 수도 있습니다. 명령:

`--output text` 옵션. 명령:

옵션. 명령:

```
aws redshift describe-clusters --output text
```

결과:

```
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
```

- 자세한 API 내용은 명령 참조DescribeClusters의 섹션을 참조하세요. AWS CLI

### `describe-default-cluster-parameters`

다음 코드 예시에서는 `describe-default-cluster-parameters`을 사용하는 방법을 보여 줍니다.

AWS CLI

기본 클러스터 설명 가져오기 ParametersThis 예제는 `redshift-1.0` 패밀리의 기본 클러스터 파라미터에 대한 설명을 반환합니다. 기본적으로 출력은 JSON 형식입니다.명령:

```
aws redshift describe-default-cluster-parameters --parameter-group-family
 redshift-1.0
```

결과:

```
{
   "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...)
      ]
   }
}
```

유효한 파라미터 그룹 패밀리 목록을 보려면 `describe-cluster-parameter-groups` 명령을 사용합니다.

`describe-cluster-parameter-groups` 명령.

명령.

- 자세한 API 내용은 명령 참조DescribeDefaultClusterParameters의 섹션을 참조하세요. AWS CLI

### `describe-event-categories`

다음 코드 예시에서는 `describe-event-categories`을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터의 이벤트 범주를 설명하려면

다음 `describe-event-categories` 예제에서는 클러스터의 이벤트 범주에 대한 세부 정보를 표시합니다.

```
aws redshift describe-event-categories \
    --source-type cluster
```

출력:

```
{
```

```
    "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"
                }
            ]
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeEventCategories의 섹션을 참조하세요. AWS CLI

### `describe-event-subscriptions`

다음 코드 예시에서는 `describe-event-subscriptions`을 사용하는 방법을 보여 줍니다.

AWS CLI

이벤트 구독을 설명하려면

다음 describe-event-subscriptions 예제에서는 지정된 구독에 대한 이벤트 알림 구독을 표시합니다.

```
aws redshift describe-event-subscriptions \
    --subscription-name mysubscription
```

출력:

```
{
    "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": []
        }
    ]
}
```

자세한 내용은 Amazon Redshift 클러스터 관리 안내서의 Amazon Redshift 이벤트 알림 구독을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeEventSubscriptions의 섹션을 참조하세요. AWS CLI

### describe-events

다음 코드 예시에서는 describe-events을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 이벤트 설명이 예제는 모든 이벤트를 반환합니다. 기본적으로 출력은 JSON 형식입니다.명령:

```
aws redshift describe-events
```

결과:

```
{
   "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"
   }
}
```

`--output text` 옵션을 사용하여 텍스트 형식으로 동일한 정보를 얻을 수도 있습니다. 명령:

`--output text` 옵션. 명령:

옵션. 명령:

```
aws redshift describe-events --output text
```

결과:

```
2013-01-22T19:17:03.640Z    myclusterparametergroup Cluster parameter group
 myclusterparametergroup has been created.       cluster-parameter-group
RESPONSEMETADATA    8e5fe765-64c9-11e2-bce3-e56f52c50e17
```

- 자세한 API 내용은 명령 참조DescribeEvents의 섹션을 참조하세요. AWS CLI

### `describe-hsm-client-certificates`

다음 코드 예시에서는 `describe-hsm-client-certificates`을 사용하는 방법을 보여 줍니다.

AWS CLI

HSM 클라이언트 인증서를 설명하려면

다음 describe-hsm-client-certificates 예제에서는 지정된 HSM 클라이언트 인증서에 대한 세부 정보를 표시합니다.

```
aws redshift describe-hsm-client-certificates \
    --hsm-client-certificate-identifier myhsmclientcert
```

출력:

```
{
    "HsmClientCertificates": [
        {
        "HsmClientCertificateIdentifier": "myhsmclientcert",
        "HsmClientCertificatePublicKey": "-----BEGIN CERTIFICATE-----\
        EXAMPLECAfICCQD6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
        VVMxCzAJBgNVBAEXAMPLERAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
        b24xFDASBgNVBAsTC0lBTSBDb25zEXAMPLEwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
        BgkqhkiG9w0BCQEWEG5vb25lQGFtYXpvbi5jb20wHhEXAMPLEDI1MjA0EXAMPLEN
        EXAMPLE0MjA0NTIxWjCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYD
        VQQHEwdTZWF0dGEXAMPLEQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25z
        b2xlMRIwEAYDVQQDEwlUZXN0Q2lsEXAMPLEdBgkqhkiG9w0BCQEWEG5vb25lQGFt
        YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIEXAMPLEMaK0dn+a4GmWIWJ
        21uUSfwfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
        rDHudUZg3qX4waLG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY67EXAMPLEE
        EXAMPLEZnzcvQAaRHhdlQWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
        nUhVVxYUntneD9EXAMPLE6q+auNKyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
        FFBjvSfpJIlJ00zbhNYS5f6GuoEDEXAMPLEBHjJnyp378OD8uTs7fLvjx79LjSTb
        NYiytVbZPQUQ5Yaxu2jXnimvw3rEXAMPLE=-----END CERTIFICATE-----\n",
        "Tags": []
        }
    ]
}
```

자세한 내용은 Amazon Redshift 클러스터 관리 안내서의 Amazon Redshift API 권한 참조를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeHsmClientCertificates의 섹션을 참조하세요. AWS CLI

### describe-hsm-configurations

다음 코드 예시에서는 describe-hsm-configurations을 사용하는 방법을 보여 줍니다.

AWS CLI

HSM 구성을 설명하려면

다음 describe-hsm-configurations 예제에서는 호출 AWS 계정에 사용 가능한 HSM 구성에 대한 세부 정보를 표시합니다.

```
aws redshift describe-hsm-configurations /
    --hsm-configuration-identifier myhsmconnection
```

출력:

```
{
    "HsmConfigurations": [
        {
            "HsmConfigurationIdentifier": "myhsmconnection",
            "Description": "My HSM connection",
            "HsmIpAddress": "192.0.2.09",
            "HsmPartitionName": "myhsmpartition",
            "Tags": []
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeHsmConfigurations의 섹션을 참조하세요. AWS CLI

### describe-logging-status

다음 코드 예시에서는 describe-logging-status을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터의 로깅 상태를 설명하려면

다음 describe-logging-status 예제에서는 쿼리 및 연결 시도와 같은 정보가 클러스터에 로깅되고 있는지 여부를 보여줍니다.

```
aws redshift describe-logging-status \
    --cluster-identifier mycluster
```

출력:

```
{
    "LoggingEnabled": false
}
```

자세한 내용은 Amazon Redshift 클러스터 관리 안내서의 데이터베이스 감사 로깅을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeLoggingStatus의 섹션을 참조하세요. AWS CLI

### describe-node-configuration-options

다음 코드 예시에서는 describe-node-configuration-options을 사용하는 방법을 보여 줍니다.

AWS CLI

노드 구성 옵션을 설명하려면

다음 describe-node-configuration-options 예제에서는 지정된 클러스터 스냅샷에 대한 노드 유형, 노드 수 및 디스크 사용량과 같은 가능한 노드 구성의 속성을 표시합니다.

```
aws redshift describe-node-configuration-options \
    --action-type restore-cluster \
    --snapshot-identifier rs:mycluster-2019-12-09-16-42-43
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 Amazon Redshift 클러스터 관리 안내서의 Amazon Redshift 예약 노드 구매를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeNodeConfigurationOptions의 섹션을 참조하세요. AWS CLI

### `describe-orderable-cluster-options`

다음 코드 예시에서는 `describe-orderable-cluster-options`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 주문 가능 클러스터 설명 OptionsThis 예제는 모든 주문 가능 클러스터 옵션에 대한 설명을 반환합니다. 기본적으로 출력은 JSON 형식입니다.명령:

```
aws redshift describe-orderable-cluster-options
```

결과:

```
{
   "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"
   }
}
```

`--output text` 옵션을 사용하여 텍스트 형식으로 동일한 정보를 얻을 수도 있습니다. 명령:

`--output text` 옵션. 명령:

옵션. 명령:

```
aws redshift describe-orderable-cluster-options --output text
```

결과:

```
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
```

- 자세한 API 내용은 명령 참조DescribeOrderableClusterOptions의 섹션을 참조하세요. AWS CLI

### `describe-reserved-node-offerings`

다음 코드 예시에서는 `describe-reserved-node-offerings`을 사용하는 방법을 보여 줍니다.

AWS CLI

예약 노드 설명 OfferingsThis 예제는 구매 가능한 모든 예약 노드 제품을 보여줍니다.명령:

```
aws redshift describe-reserved-node-offerings
```

결과:

```
{
   "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"
    }
}
```

예약된 노드 제품을 구매하려면 유효한 를 purchase-reserved-node-offering 사용하여 를 호출할 수 있습니다ReservedNodeOfferingId.

purchase-reserved-node-offering 유효한 를 사용합니다ReservedNodeOfferingId.

유효한 를 사용합니다ReservedNodeOfferingId.

ReservedNodeOfferingId.

.

- 자세한 API 내용은 명령 참조DescribeReservedNodeOfferings의 섹션을 참조하세요. AWS CLI

### describe-reserved-nodes

다음 코드 예시에서는 describe-reserved-nodes을 사용하는 방법을 보여 줍니다.

AWS CLI

예약 설명 NodesThis 예제는 구매한 예약 노드 제품을 보여줍니다.명령:

```
aws redshift describe-reserved-nodes
```

결과:

```
{
    "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"
         }
      ]
}
```

- 자세한 API 내용은 명령 참조DescribeReservedNodes의 섹션을 참조하세요. AWS CLI

### `describe-resize`

다음 코드 예시에서는 `describe-resize`을 사용하는 방법을 보여 줍니다.

AWS CLI

설명 ResizeThis 예제는 클러스터의 최신 크기 조정을 설명합니다. 요청은 유형 의 3개 노드에 대한 것이었습니다`dw.hs1.8xlarge`.명령:

```
aws redshift describe-resize --cluster-identifier mycluster
```

결과:

```
{
   "Status": "NONE",
   "TargetClusterType": "multi-node",
   "TargetNodeType": "dw.hs1.8xlarge",
   "ResponseMetadata": {
      "RequestId": "9f52b0b4-7733-11e2-aa9b-318b2909bd27"
   },
   "TargetNumberOfNodes": "3"
}
```

- 자세한 API 내용은 명령 참조DescribeResize의 섹션을 참조하세요. AWS CLI

### describe-scheduled-actions

다음 코드 예시에서는 describe-scheduled-actions을 사용하는 방법을 보여 줍니다.

AWS CLI

예약된 작업을 설명하려면

다음 describe-scheduled-actions 예제에서는 현재 예약된 작업에 대한 세부 정보를 표시합니다.

```
aws redshift describe-scheduled-actions
```

출력:

```
{
    "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"
            ]
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeScheduledActions의 섹션을 참조하세요. AWS CLI

### describe-snapshot-copy-grants

다음 코드 예시에서는 describe-snapshot-copy-grants을 사용하는 방법을 보여 줍니다.

AWS CLI

스냅샷 복사 권한 부여를 설명하려면

다음 describe-snapshot-copy-grants 예제에서는 지정된 클러스터 스냅샷 복사 권한 부여에 대한 세부 정보를 표시합니다.

```
aws redshift describe-snapshot-copy-grants \
    --snapshot-copy-grant-name mysnapshotcopygrantname
```

출력:

```
{
    "SnapshotCopyGrants": [
        {
            "SnapshotCopyGrantName": "mysnapshotcopygrantname",
            "KmsKeyId": "arn:aws:kms:us-west-2:123456789012:key/
bPxRfih3yCo8nvbEXAMPLEKEY",
            "Tags": []
        }
    ]
}
```

자세한 내용은 Amazon Redshift 클러스터 관리 안내서의 Amazon Redshift 데이터베이스 암호화를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeSnapshotCopyGrants의 섹션을 참조하세요. AWS CLI

### describe-snapshot-schedules

다음 코드 예시에서는 describe-snapshot-schedules을 사용하는 방법을 보여 줍니다.

AWS CLI

스냅샷 일정을 설명하려면

다음 describe-snapshot-schedules 예제에서는 지정된 클러스터 스냅샷 일정에 대한 세부 정보를 표시합니다.

```
aws redshift describe-snapshot-schedules \
```

```
    --cluster-identifier mycluster \
    --schedule-identifier mysnapshotschedule
```

출력:

```
{
    "SnapshotSchedules": [
        {
            "ScheduleDefinitions": [
                "rate(12 hours)"
            ],
            "ScheduleIdentifier": "mysnapshotschedule",
            "ScheduleDescription": "My schedule description",
            "Tags": [],
            "AssociatedClusterCount": 1,
            "AssociatedClusters": [
                {
                    "ClusterIdentifier": "mycluster",
                    "ScheduleAssociationState": "ACTIVE"
                }
            ]
        }
    ]
}
```

자세한 내용은 Amazon Redshift 클러스터 관리 안내서의 자동 스냅샷 일정을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeSnapshotSchedules의 섹션을 참조하세요. AWS CLI

### describe-storage

다음 코드 예시에서는 describe-storage을 사용하는 방법을 보여 줍니다.

AWS CLI

스토리지를 설명하려면

다음 describe-storage 예제에서는 계정의 백업 스토리지 및 임시 스토리지 크기에 대한 세부 정보를 보여줍니다.

```
aws redshift describe-storage
```

출력:

```
{
    "TotalBackupSizeInMegaBytes": 193149.0,
    "TotalProvisionedStorageInMegaBytes": 655360.0
}
```

자세한 내용은 Amazon Redshift 클러스터 관리 안내서의 스냅샷 스토리지 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeStorage의 섹션을 참조하세요. AWS CLI

### describe-table-restore-status

다음 코드 예시에서는 describe-table-restore-status을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터 스냅샷에서 테이블 복원 요청의 상태를 설명하려면

다음 describe-table-restore-status 예제에서는 지정된 클러스터에 대한 테이블 복원 요청에 대한 세부 정보를 표시합니다.

```
aws redshift describe-table-restore-status /
    --cluster-identifier mycluster
```

출력:

```
{
    "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"
        }
```

```
    ]
}
```

자세한 내용은 Amazon Redshift 클러스터 관리 안내서의 스냅샷에서 테이블 복원을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeTableRestoreStatus의 섹션을 참조하세요. AWS CLI

### `describe-tags`

다음 코드 예시에서는 `describe-tags`을 사용하는 방법을 보여 줍니다.

AWS CLI

태그를 설명하려면

다음 `describe-tags` 예제에서는 지정된 태그 이름 및 값과 연결된 지정된 클러스터의 리소스를 표시합니다.

```
aws redshift describe-tags \
    --resource-name arn:aws:redshift:us-west-2:123456789012:cluster:mycluster \
    --tag-keys clustertagkey \
    --tag-values clustertagvalue
```

출력:

```
{
    "TaggedResources": [
                    {
            "Tag": {
                "Key": "clustertagkey",
                "Value": "clustertagvalue"
            },
            "ResourceName": "arn:aws:redshift:us-
west-2:123456789012:cluster:mycluster",
            "ResourceType": "cluster"
        }
    ]
}
```

자세한 내용은 Amazon Redshift 클러스터 관리 안내서의 Amazon Redshift에서 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeTags의 섹션을 참조하세요. AWS CLI

### `disable-snapshot-copy`

다음 코드 예시에서는 `disable-snapshot-copy`을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터의 스냅샷 복사를 비활성화하려면

다음 `disable-snapshot-copy` 예제에서는 지정된 클러스터에 대한 스냅샷의 자동 복사본을 비활성화합니다.

```
aws redshift disable-snapshot-copy \
    --cluster-identifier mycluster
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Redshift 클러스터 관리 안내서의 다른 AWS 리전에 스냅샷 복사를 참조하세요.

- 자세한 API 내용은 명령 참조DisableSnapshotCopy의 섹션을 참조하세요. AWS CLI

### enable-snapshot-copy

다음 코드 예시에서는 enable-snapshot-copy을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터의 스냅샷 복사를 활성화하려면

다음 enable-snapshot-copy 예제에서는 지정된 클러스터에 대한 스냅샷의 자동 복사본을 활성화합니다.

```
aws redshift enable-snapshot-copy \
    --cluster-identifier mycluster \
    --destination-region us-west-1
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Redshift 클러스터 관리 안내서의 다른 AWS 리전에 스냅샷 복사를 참조하세요.

- 자세한 API 내용은 명령 참조EnableSnapshotCopy의 섹션을 참조하세요. AWS CLI

### `get-cluster-credentials`

다음 코드 예시에서는 `get-cluster-credentials`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 계정의 클러스터 자격 증명을 가져오려면

다음 `get-cluster-credentials` 예제에서는 Amazon Redshift 데이터베이스에 액세스할 수 있는 임시 보안 인증 정보를 검색합니다.

```
aws redshift get-cluster-credentials \
    --db-user adminuser --db-name dev \
    --cluster-identifier mycluster
```

출력:

```
{
    "DbUser": "IAM:adminuser",
    "DbPassword": "AMAFUyyuros/QjxPTtgzcsuQsqzIasdzJEN04aCtWDzXx1O9d6UmpkBtvEeqFly/
EXAMPLE==",
    "Expiration": "2019-12-10T17:25:05.770Z"
}
```

자세한 내용은 Amazon Redshift를 사용하여 IAM 데이터베이스 자격 증명 생성 CLI 또는 Amazon Redshift 클러스터 관리 안내서 API를 참조하세요.

- 자세한 API 내용은 명령 참조GetClusterCredentials의 섹션을 참조하세요. AWS CLI

### `get-reserved-node-exchange-offerings`

다음 코드 예시에서는 `get-reserved-node-exchange-offerings`을 사용하는 방법을 보여 줍니다.

AWS CLI

예약된 노드 교환 제품을 가져오려면

다음 get-reserved-node-exchange-offerings 예제에서는 지정된 DC1 예약 노드와 DC2 ReservedNodeOfferings 일치하는 의 배열을 검색합니다.

```
aws redshift get-reserved-node-exchange-offerings \
    --reserved-node-id 12345678-12ab-12a1-1a2a-12ab-12a12EXAMPLE
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Redshift 클러스터 관리 안내서의 를 사용하여 예약 노드 업그레이드 AWS CLI 섹션을 참조하세요.

- 자세한 API 내용은 명령 참조GetReservedNodeExchangeOfferings의 섹션을 참조하세요. AWS CLI

### modify-cluster-iam-roles

다음 코드 예시에서는 modify-cluster-iam-roles을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터의 IAM 역할을 수정하려면

다음 modify-cluster-iam-roles 예제에서는 지정된 AWS IAM 클러스터에서 지정된 역할을 제거합니다.

```
aws redshift modify-cluster-iam-roles \
    --cluster-identifier mycluster \
    --remove-iam-roles arn:aws:iam::123456789012:role/myRedshiftRole
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Redshift 클러스터 관리 안내서의 Amazon Redshift에 대한 자격 증명 기반 정책(IAM 정책) 사용을 참조하세요.

- 자세한 API 내용은 명령 참조ModifyClusterIamRoles의 섹션을 참조하세요. AWS CLI

### `modify-cluster-maintenance`

다음 코드 예시에서는 `modify-cluster-maintenance`을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터 유지 관리를 수정하려면

다음 modify-cluster-maintenance 예제에서는 지정된 클러스터의 유지 관리를 30일 지연합니다.

```
aws redshift modify-cluster-maintenance \
    --cluster-identifier mycluster \
    --defer-maintenance \
    --defer-maintenance-duration 30
```

출력:

```
{
    "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-mUdVIfFcT1B4SGhw6fyF",
                "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"
    }
}
```

자세한 내용은 Amazon Redshift 클러스터 관리 안내서의 클러스터 유지 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ModifyClusterMaintenance의 섹션을 참조하세요. AWS CLI

### `modify-cluster-parameter-group`

다음 코드 예시에서는 `modify-cluster-parameter-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

파라미터 그룹에서 파라미터 수정

다음 modify-cluster-parameter-group 예제에서는 워크로드 관리를 위해 wlm_json_configuration 파라미터를 수정합니다. 아래 표시된 JSON 내용이 포함된 파일에서 파라미터를 허용합니다.

```
aws redshift modify-cluster-parameter-group \
    --parameter-group-name myclusterparametergroup \
    --parameters file://modify_pg.json
```

modify_pg.json의 콘텐츠:

```
[
    {
        "ParameterName": "wlm_json_configuration",
        "ParameterValue": "[{\"user_group\":\"example_user_group1\",\"query_group\":
 \"example_query_group1\", \"query_concurrency\":7},{\"query_concurrency\":5}]"
    }
]
```

출력:

```
{
   "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"
   }
}
```

- 자세한 API 내용은 명령 참조ModifyClusterParameterGroup의 섹션을 참조하세요. AWS CLI

### modify-cluster-snapshot-schedule

다음 코드 예시에서는 modify-cluster-snapshot-schedule을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터 스냅샷 일정을 수정하려면

다음 modify-cluster-snapshot-schedule 예제에서는 지정된 클러스터에서 지정된 스냅샷 일정을 제거합니다.

```
aws redshift modify-cluster-snapshot-schedule \
    --cluster-identifier mycluster \
    --schedule-identifier mysnapshotschedule \
    --disassociate-schedule
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Redshift 클러스터 관리 안내서의 자동 스냅샷 일정을 참조하세요.

- 자세한 API 내용은 명령 참조ModifyClusterSnapshotSchedule의 섹션을 참조하세요. AWS CLI

### modify-cluster-snapshot

다음 코드 예시에서는 modify-cluster-snapshot을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터 스냅샷을 수정하려면

다음 modify-cluster-snapshot 예제에서는 지정된 클러스터 스냅샷의 수동 보존 기간 설정을 10일 값으로 설정합니다.

```
aws redshift modify-cluster-snapshot \
    --snapshot-identifier mycluster-2019-11-06-16-32 \
    --manual-snapshot-retention-period 10
```

출력:

```
{
    "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-b1ce17t9",
        "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"
    }
}
```

자세한 내용은 Amazon Redshift 클러스터 관리 안내서의 Amazon Redshift 스냅샷을 참조하세요.

- 자세한 API 내용은 명령 참조ModifyClusterSnapshot의 섹션을 참조하세요. AWS CLI

### `modify-cluster-subnet-group`

다음 코드 예시에서는 `modify-cluster-subnet-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터 서브넷의 서브넷 수정 GroupThis 예제는 캐시 서브넷 그룹의 서브넷 목록을 수정하는 방법을 보여줍니다. 기본적으로 출력은 JSON 형식입니다.명령:

```
aws redshift modify-cluster-subnet-group --cluster-subnet-group-name mysubnetgroup
 --subnet-ids subnet-763fdd1 subnet-ac830e9
```

결과:

```
{
   "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"
   }
}
```

- 자세한 API 내용은 명령 참조ModifyClusterSubnetGroup의 섹션을 참조하세요. AWS CLI

### `modify-cluster`

다음 코드 예시에서는 `modify-cluster`을 사용하는 방법을 보여 줍니다.

AWS CLI

보안 그룹을 ClusterThis 예제와 연결하면 클러스터 보안 그룹을 지정된 클러스터와 연결하는 방법을 보여줍니다.명령:

```
aws redshift modify-cluster --cluster-identifier mycluster --cluster-security-groups
 mysecuritygroup
```

의 유지 관리 기간을 수정하면 클러스터의 주별 기본 유지 관리 기간을 일요일 오후 11시 15분부터 월요일 오전 3시 15분까지 최소 4시간 기간으로 변경하는 방법이 ClusterThis 표시됩니다.명령:

```
aws redshift modify-cluster --cluster-identifier mycluster --preferred-maintenance-
window Sun:23:15-Mon:03:15
```

ClusterThis 예제의 마스터 암호 변경 은 클러스터의 마스터 암호를 변경하는 방법을 보여줍니다.명령:

```
aws redshift modify-cluster --cluster-identifier mycluster --master-user-password
 A1b2c3d4
```

- 자세한 API 내용은 명령 참조ModifyCluster의 섹션을 참조하세요. AWS CLI

### `modify-event-subscription`

다음 코드 예시에서는 `modify-event-subscription`을 사용하는 방법을 보여 줍니다.

AWS CLI

이벤트 구독을 수정하려면

다음 `modify-event-subscription` 예제에서는 지정된 이벤트 알림 구독을 비활성화합니다.

```
aws redshift modify-event-subscription \
    --subscription-name mysubscription \
    --no-enabled
```

출력:

```
{
    "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": []
    }
}
```

자세한 내용은 Amazon Redshift 클러스터 관리 안내서의 Amazon Redshift 이벤트 알림 구독을 참조하세요.

- 자세한 API 내용은 명령 참조ModifyEventSubscription의 섹션을 참조하세요. AWS CLI

### `modify-scheduled-action`

다음 코드 예시에서는 `modify-scheduled-action`을 사용하는 방법을 보여 줍니다.

AWS CLI

예약된 작업을 수정하려면

다음 `modify-scheduled-action` 예제에서는 지정된 기존 예약 작업에 대한 설명을 추가합니다.

```
aws redshift modify-scheduled-action \
    --scheduled-action-name myscheduledaction \
    --scheduled-action-description "My scheduled action"
```

출력:

```
{
    "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"
    ]
```

```
}
```

- 자세한 API 내용은 명령 참조ModifyScheduledAction의 섹션을 참조하세요. AWS CLI

### `modify-snapshot-copy-retention-period`

다음 코드 예시에서는 `modify-snapshot-copy-retention-period`을 사용하는 방법을 보여 줍니다.

AWS CLI

스냅샷 복사 보존 기간을 수정하려면

다음 `modify-snapshot-copy-retention-period` 예제에서는 원본 AWS 리전에서 복사된 후 대상 AWS 리전에서 지정된 클러스터의 스냅샷을 보존할 일수를 수정합니다.

```
aws redshift modify-snapshot-copy-retention-period \
    --cluster-identifier mycluster \
    --retention-period 15
```

출력:

```
{
    "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"
```

```
    }
}
```

자세한 내용은 Amazon Redshift 클러스터 관리 안내서의 스냅샷 일정 형식을 참조하세요.

- 자세한 API 내용은 명령 참조ModifySnapshotCopyRetentionPeriod의 섹션을 참조하세요. AWS CLI

### `modify-snapshot-schedule`

다음 코드 예시에서는 `modify-snapshot-schedule`을 사용하는 방법을 보여 줍니다.

AWS CLI

스냅샷 일정을 수정하려면

다음 `modify-snapshot-schedule` 예제에서는 지정된 스냅샷 일정의 속도를 10시간마다로 수정합니다.

```
aws redshift modify-snapshot-schedule \
    --schedule-identifier mysnapshotschedule \
    --schedule-definitions "rate(10 hours)"
```

출력:

```
{
    "ScheduleDefinitions": [
        "rate(10 hours)"
    ],
    "ScheduleIdentifier": "mysnapshotschedule",
    "ScheduleDescription": "My schedule description",
    "Tags": []
}
```

자세한 내용은 Amazon Redshift 클러스터 관리 안내서의 스냅샷 일정 형식을 참조하세요.

- 자세한 API 내용은 명령 참조ModifySnapshotSchedule의 섹션을 참조하세요. AWS CLI

### `purchase-reserved-node-offering`

다음 코드 예시에서는 `purchase-reserved-node-offering`을 사용하는 방법을 보여 줍니다.

AWS CLI

예약 구매 NodeThis 예제는 예약 노드 제품을 구매하는 방법을 보여줍니다. `reserved-node-offering-id` 는 `describe-reserved-node-offerings`.Command를 호출하여 가져옵니다.

```
aws redshift purchase-reserved-node-offering --reserved-node-offering-id ceb6a579-cf4c-4343-be8b-d832c45ab51c
```

결과:

```
{
   "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"
   }
}
```

- 자세한 API 내용은 명령 참조PurchaseReservedNodeOffering의 섹션을 참조하세요. AWS CLI

### `reboot-cluster`

다음 코드 예시에서는 `reboot-cluster`을 사용하는 방법을 보여 줍니다.

AWS CLI

ClusterThis 예제를 재부팅하면 클러스터가 재부팅됩니다. 기본적으로 출력은 JSON 형식입니다.명령:

```
aws redshift reboot-cluster --cluster-identifier mycluster
```

결과:

```
{
   "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"
    }
}
```

- 자세한 API 내용은 명령 참조RebootCluster의 섹션을 참조하세요. AWS CLI

### reset-cluster-parameter-group

다음 코드 예시에서는 reset-cluster-parameter-group을 사용하는 방법을 보여 줍니다.

AWS CLI

파라미터 GroupThis 예제의 파라미터 재설정은 파라미터 그룹의 모든 파라미터를 재설정하는 방법을 보여줍니다.명령:

```
aws redshift reset-cluster-parameter-group --parameter-group-name
 myclusterparametergroup --reset-all-parameters
```

- 자세한 API 내용은 명령 참조ResetClusterParameterGroup의 섹션을 참조하세요. AWS CLI

### resize-cluster

다음 코드 예시에서는 resize-cluster을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터 크기 조정

다음 resize-cluster 예제에서는 지정된 클러스터의 크기를 조정합니다.

```
aws redshift resize-cluster \
    --cluster-identifier mycluster \
    --cluster-type multi-node \
    --node-type dc2.large \
    --number-of-nodes 6 \
    --classic
```

출력:

```
{
    "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
        }
    }
}
```

자세한 내용은 Amazon Redshift 클러스터 관리 안내서의 클러스터 크기 조정을 참조하세요.

- 자세한 API 내용은 명령 참조ResizeCluster의 섹션을 참조하세요. AWS CLI

### `restore-from-cluster-snapshot`

다음 코드 예시에서는 `restore-from-cluster-snapshot`을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터 복원 SnapshotThis 예제에서 스냅샷에서 클러스터를 복원합니다.명령:

```
aws redshift restore-from-cluster-snapshot --cluster-identifier mycluster-clone --
snapshot-identifier my-snapshot-id
```

결과:

```
{
```

```
    "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"
    }
}
```

- 자세한 API 내용은 명령 참조RestoreFromClusterSnapshot의 섹션을 참조하세요. AWS CLI

### `restore-table-from-cluster-snapshot`

다음 코드 예시에서는 `restore-table-from-cluster-snapshot`을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터 스냅샷에서 테이블을 복원하려면

다음 `restore-table-from-cluster-snapshot` 예제에서는 지정된 클러스터 스냅샷의 지정된 테이블에서 새 테이블을 생성합니다.

```
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
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Redshift 클러스터 관리 안내서의 스냅샷에서 테이블 복원을 참조하세요.

- 자세한 API 내용은 명령 참조RestoreTableFromClusterSnapshot의 섹션을 참조하세요. AWS CLI

### `revoke-cluster-security-group-ingress`

다음 코드 예시에서는 `revoke-cluster-security-group-ingress`을 사용하는 방법을 보여 줍니다.

AWS CLI

EC2 보안 GroupThis 예제에서 액세스 취소는 이름이 지정된 Amazon EC2 보안 그룹에 대한 액세스를 취소합니다.명령:

```
aws redshift revoke-cluster-security-group-ingress --cluster-security-group-name
 mysecuritygroup --ec2-security-group-name myec2securitygroup --ec2-security-group-
owner-id 123445677890
```

CIDR rangeThis 예제에 대한 액세스를 취소하면 CIDR 범위에 대한 액세스가 취소됩니다.명령:

```
aws redshift revoke-cluster-security-group-ingress --cluster-security-group-name
 mysecuritygroup --cidrip 192.168.100.100/32
```

- 자세한 API 내용은 명령 참조RevokeClusterSecurityGroupIngress의 섹션을 참조하세요. AWS CLI

### `revoke-snapshot-access`

다음 코드 예시에서는 `revoke-snapshot-access`을 사용하는 방법을 보여 줍니다.

AWS CLI

SnapshotThis 예제를 복원하기 위한 AWS 계정 권한 부여를 취소하면 스냅샷을 복원444455556666하기 위한 AWS 계정 권한이 취소됩니다`my-snapshot-id`. 기본적으로 출력은 JSON 형식입니다.명령:

```
aws redshift revoke-snapshot-access --snapshot-id my-snapshot-id --account-with-
restore-access 444455556666
```

결과:

```
{
   "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"
   }
}
```

- 자세한 API 내용은 명령 참조RevokeSnapshotAccess의 섹션을 참조하세요. AWS CLI

### rotate-encryption-key

다음 코드 예시에서는 rotate-encryption-key을 사용하는 방법을 보여 줍니다.

AWS CLI

클러스터의 암호화 키를 교체하려면

다음 rotate-encryption-key 예제에서는 지정된 클러스터의 암호화 키를 교체합니다.

```
aws redshift rotate-encryption-key \
    --cluster-identifier mycluster
```

출력:

```
{
    "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"
    }
```

```
}
```

자세한 내용은 Amazon Redshift 클러스터 관리 안내서의 Amazon Redshift 데이터베이스 암호화를 참조하세요.

- 자세한 API 내용은 명령 참조RotateEncryptionKey의 섹션을 참조하세요. AWS CLI

## 를 사용한 Amazon Rekognition 예제 AWS CLI

다음 코드 예제에서는 Amazon Rekognition 과 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `compare-faces`

다음 코드 예시에서는 `compare-faces`을 사용하는 방법을 보여 줍니다.

자세한 내용은 이미지 내 얼굴 비교를 참조하세요.

AWS CLI

두 이미지에서 얼굴을 비교하는 방법

다음 `compare-faces` 명령은 Amazon S3 버킷에 저장된 두 이미지에서 얼굴을 비교합니다.

```
aws rekognition compare-faces \
    --source-image '{"S3Object":{"Bucket":"MyImageS3Bucket","Name":"source.jpg"}}' \
    --target-image '{"S3Object":{"Bucket":"MyImageS3Bucket","Name":"target.jpg"}}'
```

출력:


```
{
    "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
    }
}
```

자세한 내용은 Amazon Rekognition 개발자 안내서의 이미지에 있는 얼굴 비교를 참조하세요.

- 자세한 API 내용은 명령 참조CompareFaces의 섹션을 참조하세요. AWS CLI

### `create-collection`

다음 코드 예시에서는 `create-collection`을 사용하는 방법을 보여 줍니다.

자세한 내용은 컬렉션 생성을 참조하세요.

AWS CLI

모음을 생성하는 방법

다음 `create-collection` 명령을 실행하면 지정된 이름의 모음이 생성됩니다.

```
aws rekognition create-collection \
    --collection-id "MyCollection"
```

출력:

```
{
```

```
    "CollectionArn": "aws:rekognition:us-west-2:123456789012:collection/
MyCollection",
    "FaceModelVersion": "4.0",
    "StatusCode": 200
}
```

자세한 내용은 Amazon Rekognition 개발자 안내서의 모음 만들기를 참조하세요.

- 자세한 API 내용은 명령 참조CreateCollection의 섹션을 참조하세요. AWS CLI

### create-stream-processor

다음 코드 예시에서는 create-stream-processor을 사용하는 방법을 보여 줍니다.

AWS CLI

새 스트림 프로세서를 생성하려면

다음 create-stream-processor 예제에서는 지정된 구성을 사용하여 새 스트림 프로세서를 생성합니다.

```
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}}'
```

출력:

```
{
    "StreamProcessorArn": "arn:aws:rekognition:us-
west-2:123456789012:streamprocessor/my-stream-processor"
}
```

자세한 내용은 Amazon Rekognition 개발자 안내서의 스트리밍 비디오 작업을 참조하세요.

- 자세한 API 내용은 명령 참조CreateStreamProcessor의 섹션을 참조하세요. AWS CLI

### delete-collection

다음 코드 예시에서는 delete-collection을 사용하는 방법을 보여 줍니다.

자세한 내용은 컬렉션을 삭제를 참조하세요.

AWS CLI

모음을 삭제하는 방법

다음 delete-collection 명령은 지정된 모음을 삭제합니다.

```
aws rekognition delete-collection \
    --collection-id MyCollection
```

출력:

```
{
    "StatusCode": 200
}
```

자세한 내용은 Amazon Rekognition 개발자 안내서의 모음 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteCollection의 섹션을 참조하세요. AWS CLI

### delete-faces

다음 코드 예시에서는 delete-faces을 사용하는 방법을 보여 줍니다.

자세한 내용은 컬렉션에서 얼굴 삭제를 참조하십시오.

AWS CLI

모음에서 얼굴을 삭제하는 방법

다음 delete-faces 명령은 모음에서 지정된 얼굴을 삭제합니다.

```
aws rekognition delete-faces \
    --collection-id MyCollection
    --face-ids '["0040279c-0178-436e-b70a-e61b074e96b0"]'
```

출력:

```
{
    "DeletedFaces": [
        "0040279c-0178-436e-b70a-e61b074e96b0"
    ]
}
```

자세한 내용은 Amazon Rekognition 개발자 안내서의 모음에서 얼굴 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteFaces의 섹션을 참조하세요. AWS CLI

### delete-stream-processor

다음 코드 예시에서는 delete-stream-processor을 사용하는 방법을 보여 줍니다.

AWS CLI

스트림 프로세서를 삭제하려면

다음 delete-stream-processor 명령은 지정된 스트림 프로세서를 삭제합니다.

```
aws rekognition delete-stream-processor \
    --name my-stream-processor
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Rekognition 개발자 안내서의 스트리밍 비디오 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteStreamProcessor의 섹션을 참조하세요. AWS CLI

### describe-collection

다음 코드 예시에서는 describe-collection을 사용하는 방법을 보여 줍니다.

자세한 내용은 컬렉션 설명을 참조하세요.

AWS CLI

모음을 설명하는 방법

다음 describe-collection 예시에서는 지정된 모음의 세부 정보를 표시합니다.

```
aws rekognition describe-collection \
    --collection-id MyCollection
```

출력:

```
{
    "FaceCount": 200,
    "CreationTimestamp": 1569444828.274,
    "CollectionARN": "arn:aws:rekognition:us-west-2:123456789012:collection/
MyCollection",
    "FaceModelVersion": "4.0"
}
```

자세한 내용은 Amazon Rekognition 개발자 안내서의 모음 설명을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeCollection의 섹션을 참조하세요. AWS CLI

### describe-stream-processor

다음 코드 예시에서는 describe-stream-processor을 사용하는 방법을 보여 줍니다.

AWS CLI

스트림 프로세서에 대한 정보를 가져오려면

다음 describe-stream-processor 명령은 지정된 스트림 프로세서에 대한 세부 정보를 표시합니다.

```
aws rekognition describe-stream-processor \
    --name my-stream-processor
```

출력:

```
{
    "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-processpr",
```

```
    "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
}
```

자세한 내용은 Amazon Rekognition 개발자 안내서의 스트리밍 비디오 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeStreamProcessor의 섹션을 참조하세요. AWS CLI

### `detect-faces`

다음 코드 예시에서는 `detect-faces`을 사용하는 방법을 보여 줍니다.

자세한 내용은 이미지에서 얼굴 감지를 참조하십시오.

AWS CLI

이미지에서 얼굴을 감지하는 방법

다음 `detect-faces` 명령은 Amazon S3 버킷에 저장된 지정된 이미지에서 얼굴을 감지합니다.

```
aws rekognition detect-faces \
    --image '{"S3Object":{"Bucket":"MyImageS3Bucket","Name":"MyFriend.jpg"}}' \
    --attributes "ALL"
```

출력:

```
{
    "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
            }
        }
    ]
}
```

자세한 내용은 Amazon Rekognition 개발자 안내서의 이미지에서 얼굴 감지를 참조하세요.

- 자세한 API 내용은 명령 참조DetectFaces의 섹션을 참조하세요. AWS CLI

### `detect-labels`

다음 코드 예시에서는 `detect-labels`을 사용하는 방법을 보여 줍니다.

자세한 내용은 이미지에서 레이블 감지를 참조하세요.

AWS CLI

이미지에서 레이블을 감지하는 방법

다음 `detect-labels` 예시에서는 Amazon S3 버킷에 저장된 이미지에서 장면과 객체를 감지합니다.

```
aws rekognition detect-labels \
    --image '{"S3Object":{"Bucket":"bucket","Name":"image"}}'
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon Rekognition 개발자 안내서의 이미지에서 레이블 감지를 참조하세요.

- 자세한 API 내용은 명령 참조DetectLabels의 섹션을 참조하세요. AWS CLI

### `detect-moderation-labels`

다음 코드 예시에서는 `detect-moderation-labels`을 사용하는 방법을 보여 줍니다.

자세한 내용은 부적절한 이미지 감지를 참조하세요.

AWS CLI

이미지에서 안전하지 않은 콘텐츠를 감지하는 방법

다음 `detect-moderation-labels` 명령은 Amazon S3 버킷에 저장된 지정된 이미지에서 안전하지 않은 콘텐츠를 감지합니다.

```
aws rekognition detect-moderation-labels \
    --image "S3Object={Bucket=MyImageS3Bucket,Name=gun.jpg}"
```

출력:

```
{
    "ModerationModelVersion": "3.0",
    "ModerationLabels": [
        {
            "Confidence": 97.29618072509766,
            "ParentName": "Violence",
            "Name": "Weapon Violence"
        },
        {
            "Confidence": 97.29618072509766,
            "ParentName": "",
            "Name": "Violence"
        }
    ]
}
```

자세한 내용은 Amazon Rekognition 개발자 안내서의 안전하지 않은 이미지 감지를 참조하세요.

- 자세한 API 내용은 명령 참조DetectModerationLabels의 섹션을 참조하세요. AWS CLI

### `detect-text`

다음 코드 예시에서는 `detect-text`을 사용하는 방법을 보여 줍니다.

자세한 내용은 이미지에서 텍스트 감지를 참조하세요.

AWS CLI

이미지에서 텍스트를 감지하는 방법

다음 `detect-text` 명령은 지정된 이미지에서 텍스트를 감지합니다.

```
aws rekognition detect-text \
    --image '{"S3Object":{"Bucket":"MyImageS3Bucket","Name":"ExamplePicture.jpg"}}'
```

출력:

```
{
    "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
        }
    ]
}
```


- 자세한 API 내용은 명령 참조DetectText의 섹션을 참조하세요. AWS CLI

### `disassociate-faces`

다음 코드 예시에서는 `disassociate-faces`을 사용하는 방법을 보여 줍니다.

AWS CLI

```
aws rekognition disassociate-faces --face-ids list-of-face-ids
  --user-id user-id --collection-id collection-name --region region-name
```

- 자세한 API 내용은 명령 참조DisassociateFaces의 섹션을 참조하세요. AWS CLI

### get-celebrity-info

다음 코드 예시에서는 get-celebrity-info을 사용하는 방법을 보여 줍니다.

AWS CLI

유명 인사에 대한 정보를 가져오는 방법

다음 get-celebrity-info 명령은 지정된 유명 인사에 대한 정보를 표시합니다. id 파라미터는 이전 recognize-celebrities 직접 호출에서 가져온 것입니다.

```
aws rekognition get-celebrity-info --id nnnnnnn
```

출력:

```
{
    "Name": "Celeb A",
    "Urls": [
        "www.imdb.com/name/aaaaaaaaa"
    ]
}
```

자세한 내용은 Amazon Rekognition 개발자 안내서의 유명 인사에 대한 정보 얻기를 참조하세요.

- 자세한 API 내용은 명령 참조GetCelebrityInfo의 섹션을 참조하세요. AWS CLI

### get-celebrity-recognition

다음 코드 예시에서는 get-celebrity-recognition을 사용하는 방법을 보여 줍니다.

AWS CLI

유명인 인식 작업의 결과를 얻으려면

다음 get-celebrity-recognition 명령은 이전에 를 호출하여 시작한 유명인 인식 작업의 결과를 표시합니다start-celebrity-recognition.

```
aws rekognition get-celebrity-recognition  \
    --job-id 1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef
```

출력:

```
{
    "NextToken": "3D01ClxlCiT31VsRDkAO3IybLb/h5AtDWSGuhYi
+N1FIJwwPtAkuKzDhL2rV3GcwmNt77+12",
    "Celebrities": [
        {
            "Timestamp": 0,
            "Celebrity": {
                "Confidence": 96.0,
                "Face": {
                    "BoundingBox": {
                        "Width": 0.70333331823349,
                        "Top": 0.16750000417232513,
                        "Left": 0.19555555284023285,
                        "Height": 0.3956249952316284
                    },
                    "Landmarks": [
                        {
                            "Y": 0.31031012535095215,
                            "X": 0.441436767578125,
                            "Type": "eyeLeft"
                        },
                        {
                            "Y": 0.3081788718700409,
                            "X": 0.6437258720397949,
                            "Type": "eyeRight"
                        },
                        {
                            "Y": 0.39542075991630554,
                            "X": 0.5572493076324463,
                            "Type": "nose"
                        },
                        {
                            "Y": 0.4597957134246826,
                            "X": 0.4579732120037079,
                            "Type": "mouthLeft"
                        },
                        {
                            "Y": 0.45688048005104065,
```

```
                        "X": 0.6349081993103027,
                        "Type": "mouthRight"
                    }
                ],
                "Pose": {
                    "Yaw": 8.943398475646973,
                    "Roll": -2.0309247970581055,
                    "Pitch": -0.5674862861633301
                },
                "Quality": {
                    "Sharpness": 99.40211486816406,
                    "Brightness": 89.47132110595703
                },
                "Confidence": 99.99861145019531
            },
            "Name": "CelebrityA",
            "Urls": [
                "www.imdb.com/name/111111111"
            ],
            "Id": "nnnnnn"
        }
    },
    {
        "Timestamp": 467,
        "Celebrity": {
            "Confidence": 99.0,
            "Face": {
                "BoundingBox": {
                    "Width": 0.6877777576446533,
                    "Top": 0.18437500298023224,
                    "Left": 0.20555555820465088,
                    "Height": 0.3868750035762787
                },
                "Landmarks": [
                    {
                        "Y": 0.31895750761032104,
                        "X": 0.4411413371562958,
                        "Type": "eyeLeft"
                    },
                    {
                        "Y": 0.3140959143638611,
                        "X": 0.6523157954216003,
                        "Type": "eyeRight"
                    },
```

```
                        {
                            "Y": 0.4016456604003906,
                            "X": 0.5682755708694458,
                            "Type": "nose"
                        },
                        {
                            "Y": 0.46894142031669617,
                            "X": 0.4597797095775604,
                            "Type": "mouthLeft"
                        },
                        {
                            "Y": 0.46971091628074646,
                            "X": 0.6286435127258301,
                            "Type": "mouthRight"
                        }
                    ],
                    "Pose": {
                        "Yaw": 10.433465957641602,
                        "Roll": -3.347442388534546,
                        "Pitch": 1.3709543943405151
                    },
                    "Quality": {
                        "Sharpness": 99.5531005859375,
                        "Brightness": 88.5764389038086
                    },
                    "Confidence": 99.99148559570312
                },
                "Name": "Jane Celebrity",
                "Urls": [
                    "www.imdb.com/name/111111111"
                ],
                "Id": "nnnnnn"
            }
        }
    ],
    "JobStatus": "SUCCEEDED",
    "VideoMetadata": {
        "Format": "QuickTime / MOV",
        "FrameRate": 29.978118896484375,
        "Codec": "h264",
        "DurationMillis": 4570,
        "FrameHeight": 1920,
        "FrameWidth": 1080
    }
```

```
}
```

자세한 내용은 Amazon Rekognition 개발자 안내서의 저장된 비디오에서 유명 인사 인식 섹션을 참조하세요.

- 자세한 API 내용은 명령 참조GetCelebrityRecognition의 섹션을 참조하세요. AWS CLI

### `get-content-moderation`

다음 코드 예시에서는 `get-content-moderation`을 사용하는 방법을 보여 줍니다.

AWS CLI

안전하지 않은 콘텐츠 작업의 결과를 얻으려면

다음 `get-content-moderation` 명령은 이전에 를 호출하여 시작한 안전하지 않은 콘텐츠 작업의 결과를 표시합니다`start-content-moderation`.

```
aws rekognition get-content-moderation \
    --job-id 1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef
```

출력:

```
{
    "NextToken": "dlhcKMHMzpCBGFukz6IO3JMcWiJAamCVhXHt3r6b4b5Tfbyw3q7o+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
    }
}
```

자세한 내용은 Amazon Rekognition 개발자 안내서의 안전하지 않은 저장 비디오 감지를 참조하세요.

- 자세한 API 내용은 명령 참조GetContentModeration의 섹션을 참조하세요. AWS CLI

### get-face-detection

다음 코드 예시에서는 get-face-detection을 사용하는 방법을 보여 줍니다.

AWS CLI

얼굴 감지 작업의 결과를 가져오려면

다음 get-face-detection 명령은 를 호출하여 이전에 시작한 얼굴 감지 작업의 결과를 표시합니다start-face-detection.

```
aws rekognition get-face-detection \
    --job-id 1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef
```

출력:

```
{
    "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":
 "OzL223pDKy9116O/02KXRqFIEAwxjy4PkgYcm3hSo0rdysbXg5Ex0eFgTGEj0ADEac6S037U",
    "JobStatus": "SUCCEEDED",
    "VideoMetadata": {
        "Format": "QuickTime / MOV",
        "FrameRate": 29.970617294311523,
        "Codec": "h264",
        "DurationMillis": 6806,
        "FrameHeight": 1080,
        "FrameWidth": 1920
    }
}
```

자세한 내용은 Amazon Rekognition 개발자 안내서의 저장된 비디오에서 얼굴 감지를 참조하세요.

- 자세한 API 내용은 명령 참조GetFaceDetection의 섹션을 참조하세요. AWS CLI

### get-face-search

다음 코드 예시에서는 get-face-search을 사용하는 방법을 보여 줍니다.

AWS CLI

얼굴 검색 작업의 결과를 가져오려면

다음 get-face-search 명령은 를 호출하여 이전에 시작한 얼굴 검색 작업의 결과를 표시합니다start-face-search.

```
aws rekognition get-face-search  \
    --job-id 1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef
```

출력:

```
{
    "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": "5bkgcezyuaqhtWk3C8OTW6cjRghrwV9XDMivm5B3MXm+Lv6G+L+GejyFHPhoNa/
ldXIC4c/d",
    "JobStatus": "SUCCEEDED",
    "VideoMetadata": {
        "Format": "QuickTime / MOV",
        "FrameRate": 29.970617294311523,
        "Codec": "h264",
        "DurationMillis": 6806,
        "FrameHeight": 1080,
        "FrameWidth": 1920
    }
}
```

자세한 내용은 Amazon Rekognition 개발자 안내서의 얼굴에 대한 저장된 비디오 검색을 참조하세요.

- 자세한 API 내용은 명령 참조GetFaceSearch의 섹션을 참조하세요. AWS CLI

### get-label-detection

다음 코드 예시에서는 get-label-detection을 사용하는 방법을 보여 줍니다.

AWS CLI

객체 및 장면 감지 작업의 결과를 가져오려면

다음 get-label-detection 명령은 를 호출하여 이전에 시작한 객체 및 장면 감지 작업의 결과를 표시합니다start-label-detection.

```
aws rekognition get-label-detection  \
    --job-id 1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef
```

출력:

```
{
    "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/
CpObenspRfs/5FBQFs4X7G"
}
```

자세한 내용은 Amazon Rekognition 개발자 안내서의 비디오에서 레이블 감지를 참조하세요.

- 자세한 API 내용은 명령 참조GetLabelDetection의 섹션을 참조하세요. AWS CLI

### get-person-tracking

다음 코드 예시에서는 get-person-tracking을 사용하는 방법을 보여 줍니다.

AWS CLI

인력 경로 지정 작업의 결과를 가져오려면

다음 get-person-tracking 명령은 이전에 를 호출하여 시작한 사람 경로 지정 작업의 결과를 표시합니다start-person-tracking.

```
aws rekognition get-person-tracking  \
    --job-id 1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef
```

출력:


```
{
    "Persons": [
        {
            "Timestamp": 500,
            "Person": {
                "BoundingBox": {
                    "Width": 0.4151041805744171,
                    "Top": 0.07870370149612427,
                    "Left": 0.0,
                    "Height": 0.9212962985038757
                },
                "Index": 0
            }
        },
        {
            "Timestamp": 567,
            "Person": {
                "BoundingBox": {
                    "Width": 0.4755208194255829,
                    "Top": 0.07777778059244156,
                    "Left": 0.0,
                    "Height": 0.9194444417953491
                },
                "Index": 0
            }
        }
    ],
    "NextToken": "D/vRIYNyhG79ugdta3f+8cRg9oSRo
+HigGOuxRiYpTn0ExnqTi1CJektVAc4HrAXDv25eHYk",
    "JobStatus": "SUCCEEDED",
    "VideoMetadata": {
        "Format": "QuickTime / MOV",
        "FrameRate": 29.970617294311523,
        "Codec": "h264",
        "DurationMillis": 6806,
        "FrameHeight": 1080,
        "FrameWidth": 1920
    }
}
```


자세한 내용은 Amazon Rekognition 개발자 안내서의 People Pathing을 참조하세요.

- 자세한 API 내용은 명령 참조GetPersonTracking의 섹션을 참조하세요. AWS CLI

### index-faces

다음 코드 예시에서는 index-faces을 사용하는 방법을 보여 줍니다.

자세한 내용은 컬렉션에 얼굴 추가를 참조하십시오.

AWS CLI

모음에 얼굴을 추가하는 방법

다음 index-faces 명령은 이미지에서 찾은 얼굴을 지정된 모음에 추가합니다.

```
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"
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon Rekognition 개발자 안내서의 모음에 얼굴 추가를 참조하세요.

- 자세한 API 내용은 명령 참조IndexFaces의 섹션을 참조하세요. AWS CLI

### `list-collections`

다음 코드 예시에서는 `list-collections`을 사용하는 방법을 보여 줍니다.

자세한 내용은 컬렉션 나열을 참조하세요.

AWS CLI

사용 가능한 모음을 나열하는 방법

다음 `list-collections` 명령은 AWS 계정에서 사용 가능한 컬렉션을 나열합니다.

```
aws rekognition list-collections
```

출력:

```
{
```

```
    "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"
    ]
}
```

자세한 내용은 Amazon Rekognition 개발자 안내서의 모음 나열을 참조하세요.

- 자세한 API 내용은 명령 참조ListCollections의 섹션을 참조하세요. AWS CLI

### `list-faces`

다음 코드 예시에서는 `list-faces`을 사용하는 방법을 보여 줍니다.

자세한 내용은 컬렉션에서 얼굴 나열을 참조하세요.

AWS CLI

모음에 있는 얼굴을 나열하는 방법

다음 `list-faces` 명령은 지정된 모음에 있는 얼굴을 나열합니다.

```
aws rekognition list-faces \
```

```
    --collection-id MyCollection
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Rekognition 개발자 안내서의 모음에 얼굴 나열을 참조하세요.

- 자세한 API 내용은 명령 참조ListFaces의 섹션을 참조하세요. AWS CLI

### `list-stream-processors`

다음 코드 예시에서는 `list-stream-processors`을 사용하는 방법을 보여 줍니다.

AWS CLI

계정의 스트림 프로세서를 나열하려면

다음 `list-stream-processors` 명령은 계정의 스트림 프로세서와 각 의 상태를 나열합니다.

```
aws rekognition list-stream-processors
```

출력:

```
{
    "StreamProcessors": [
        {
            "Status": "STOPPED",
            "Name": "my-stream-processor"
        }
    ]
}
```

자세한 내용은 Amazon Rekognition 개발자 안내서의 스트리밍 비디오 작업을 참조하세요.

- 자세한 API 내용은 명령 참조ListStreamProcessors의 섹션을 참조하세요. AWS CLI

### `recognize-celebrities`

다음 코드 예시에서는 `recognize-celebrities`을 사용하는 방법을 보여 줍니다.

자세한 내용은 유명 인사 인식을 참조하세요.

AWS CLI

이미지에서 유명 인사를 인식하는 방법

다음 `recognize-celebrities` 명령은 Amazon S3 버킷에 저장된 지정된 이미지에서 유명 인사를 인식합니다.

```
aws rekognition recognize-celebrities \
    --image "S3Object={Bucket=MyImageS3Bucket,Name=moviestars.jpg}"
```

출력:

```
{
    "UnrecognizedFaces": [
        {
            "BoundingBox": {
                "Width": 0.14416666328907013,
                "Top": 0.07777778059244156,
                "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/bbbbbbbbb"
            ],
            "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": "4444444"
        }
    ]
}
```

자세한 내용은 Amazon Rekognition 개발자 안내서의 이미지 속 유명 인사 인식을 참조하세요.

- 자세한 API 내용은 명령 참조RecognizeCelebrities의 섹션을 참조하세요. AWS CLI

### search-faces-by-image

다음 코드 예시에서는 search-faces-by-image을 사용하는 방법을 보여 줍니다.

자세한 내용은 얼굴(이미지) 검색을 참조하세요.

AWS CLI

이미지에서 가장 큰 얼굴과 일치하는 얼굴을 모음에서 검색하는 방법

다음 search-faces-by-image 명령은 지정된 이미지에서 가장 큰 얼굴과 일치하는 얼굴을 모음에서 검색합니다.

```
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"
}
```

자세한 내용은 Amazon Rekognition 개발자 안내서의 이미지를 사용하여 얼굴 검색을 참조하세요.

- 자세한 API 내용은 명령 참조SearchFacesByImage의 섹션을 참조하세요. AWS CLI

### `search-faces`

다음 코드 예시에서는 `search-faces`을 사용하는 방법을 보여 줍니다.

자세한 내용은 얼굴 검색(face ID)을 참조하세요.

AWS CLI

모음에서 얼굴 ID와 일치하는 얼굴을 검색하는 방법

다음 `search-faces` 명령은 모음에서 지정된 얼굴 ID와 일치하는 얼굴을 검색합니다.

```
aws rekognition search-faces \
    --face-id 8d3cfc70-4ba8-4b36-9644-90fba29c2dac \
    --collection-id MyCollection
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 Amazon Rekognition 개발자 안내서의 얼굴 ID를 사용하여 얼굴 검색을 참조하세요.

- 자세한 API 내용은 명령 참조SearchFaces의 섹션을 참조하세요. AWS CLI

### `start-celebrity-recognition`

다음 코드 예시에서는 `start-celebrity-recognition`을 사용하는 방법을 보여 줍니다.

AWS CLI

저장된 비디오에서 유명 인사의 인식을 시작하려면

다음 `start-celebrity-recognition` 명령은 Amazon S3 버킷에 저장된 지정된 비디오 파일에서 유명 인사를 찾는 작업을 시작합니다.

```
aws rekognition start-celebrity-recognition \
    --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}"
```

출력:

```
{
    "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"
}
```

자세한 내용은 Amazon Rekognition 개발자 안내서의 저장된 비디오에서 유명 인사 인식 섹션을 참조하세요.

- 자세한 API 내용은 명령 참조StartCelebrityRecognition의 섹션을 참조하세요. AWS CLI

### `start-content-moderation`

다음 코드 예시에서는 `start-content-moderation`을 사용하는 방법을 보여 줍니다.

AWS CLI

저장된 비디오에서 안전하지 않은 콘텐츠 인식을 시작하려면

다음 `start-content-moderation` 명령은 Amazon S3 버킷에 저장된 지정된 비디오 파일에서 안전하지 않은 콘텐츠를 감지하는 작업을 시작합니다.

```
aws rekognition start-content-moderation \
    --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}"
```

출력:

```
{
    "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"
}
```

자세한 내용은 Amazon Rekognition 개발자 안내서의 안전하지 않은 저장 비디오 감지를 참조하세요.

- 자세한 API 내용은 명령 참조StartContentModeration의 섹션을 참조하세요. AWS CLI

### `start-face-detection`

다음 코드 예시에서는 `start-face-detection`을 사용하는 방법을 보여 줍니다.

AWS CLI

비디오에서 얼굴을 감지하려면

다음 `start-face-detection` 명령은 Amazon S3 버킷에 저장된 지정된 비디오 파일의 얼굴을 감지하는 작업을 시작합니다.

```
aws rekognition start-face-detection
```

```
    --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}"
```

출력:

```
{
    "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"
}
```

자세한 내용은 Amazon Rekognition 개발자 안내서의 저장된 비디오에서 얼굴 감지를 참조하세요.

- 자세한 API 내용은 명령 참조StartFaceDetection의 섹션을 참조하세요. AWS CLI

### start-face-search

다음 코드 예시에서는 start-face-search을 사용하는 방법을 보여 줍니다.

AWS CLI

비디오에서 감지된 얼굴과 일치하는 컬렉션의 얼굴을 검색하려면

다음 start-face-search 명령은 Amazon S3 버킷의 지정된 비디오 파일에서 감지된 얼굴과 일치하는 컬렉션에서 얼굴을 검색하는 작업을 시작합니다.

```
aws rekognition start-face-search \
    --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}" \
    --collection-id collection
```

출력:

```
{
    "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"
}
```

자세한 내용은 Amazon Rekognition 개발자 안내서의 얼굴에 대한 저장된 비디오 검색을 참조하세요.

- 자세한 API 내용은 명령 참조StartFaceSearch의 섹션을 참조하세요. AWS CLI

### start-label-detection

다음 코드 예시에서는 start-label-detection을 사용하는 방법을 보여 줍니다.

AWS CLI

비디오에서 객체 및 장면을 감지하려면

다음 `start-label-detection` 명령은 Amazon S3 버킷에 저장된 지정된 비디오 파일의 객체 및 장면을 감지하는 작업을 시작합니다.

```
aws rekognition start-label-detection \
    --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}"
```

출력:

```
{
    "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"
}
```

자세한 내용은 Amazon Rekognition 개발자 안내서의 비디오에서 레이블 감지를 참조하세요.

- 자세한 API 내용은 명령 참조StartLabelDetection의 섹션을 참조하세요. AWS CLI

### `start-person-tracking`

다음 코드 예시에서는 `start-person-tracking`을 사용하는 방법을 보여 줍니다.

AWS CLI

저장된 비디오에서 사람들의 경로를 시작하려면

다음 `start-person-tracking` 명령은 사용자가 Amazon S3 버킷에 저장된 지정된 비디오 Fiel에서 취하는 경로를 추적하는 작업을 시작합니다.

```
aws rekognition start-person-tracking \
    --video "S3Object={Bucket=MyVideoS3Bucket,Name=MyVideoFile.mpg}"
```

출력:

```
{
    "JobId": "1234567890abcdef1234567890abcdef1234567890abcdef1234567890abcdef"
}
```

자세한 내용은 Amazon Rekognition 개발자 안내서의 People Pathing을 참조하세요.

- 자세한 API 내용은 명령 참조StartPersonTracking의 섹션을 참조하세요. AWS CLI

### start-stream-processor

다음 코드 예시에서는 start-stream-processor을 사용하는 방법을 보여 줍니다.

AWS CLI

스트림 프로세서를 시작하려면

다음 start-stream-processor 명령은 지정된 비디오 스트림 프로세서를 시작합니다.

```
aws rekognition start-stream-processor \
    --name my-stream-processor
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Rekognition 개발자 안내서의 스트리밍 비디오 작업을 참조하세요.

- 자세한 API 내용은 명령 참조StartStreamProcessor의 섹션을 참조하세요. AWS CLI

### stop-stream-processor

다음 코드 예시에서는 stop-stream-processor을 사용하는 방법을 보여 줍니다.

AWS CLI

실행 중인 스트림 프로세서를 중지하려면

다음 stop-stream-processor 명령은 지정된 실행 스트림 프로세서를 중지합니다.

```
aws rekognition stop-stream-processor \
    --name my-stream-processor
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Rekognition 개발자 안내서의 스트리밍 비디오 작업을 참조하세요.

- 자세한 API 내용은 명령 참조StopStreamProcessor의 섹션을 참조하세요. AWS CLI

# AWS RAM 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS RAM.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `accept-resource-share-invitation`

다음 코드 예시에서는 `accept-resource-share-invitation`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 공유 초대를 수락하려면

다음 `accept-resource-share-invitation` 예제에서는 지정된 리소스 공유 초대를 수락합니다. 초대된 계정의 보안 주체는 공유의 리소스를 즉시 사용할 수 있습니다.

```
aws ram accept-resource-share-invitation \
    --resource-share-invitation-arn arn:aws:ram:us-west-2:111111111111:resource-share-invitation/1e3477be-4a95-46b4-bbe0-c4001EXAMPLE
```

출력:

```
{
    "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"
    }
}
```

- 자세한 API 내용은 명령 참조AcceptResourceShareInvitation의 섹션을 참조하세요. AWS CLI

### associate-resource-share-permission

다음 코드 예시에서는 associate-resource-share-permission을 사용하는 방법을 보여 줍니다.

AWS CLI

RAM 관리형 권한을 리소스 공유에 연결하려면

다음 associate-resource-share-permission 예제에서는 관련 리소스 유형에 대한 기존 관리형 권한을 지정된 관리형 권한으로 바꿉니다. 관련 리소스 유형의 모든 리소스에 대한 액세스에는 새 권한이 적용됩니다.

```
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
```

출력:

```
{
    "returnValue": true
}
```

- 자세한 API 내용은 명령 참조AssociateResourceSharePermission의 섹션을 참조하세요. AWS CLI

### associate-resource-share

다음 코드 예시에서는 associate-resource-share을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 리소스를 리소스 공유에 연결하려면

다음 associate-resource-share 예제에서는 지정된 리소스 공유에 라이선스 구성을 추가합니다.

```
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
```

출력:

```
{
    "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
        }
    ]
}
```

예제 2: 보안 주체를 리소스 공유에 연결하려면

다음 associate-resource-share 예제에서는 지정된 리소스 공유에 대한 액세스 권한을 지정된 조직 단위의 모든 계정에 부여합니다.

```
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
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조AssociateResourceShare의 섹션을 참조하세요. AWS CLI

### `create-resource-share`

다음 코드 예시에서는 `create-resource-share`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 리소스 공유 생성

다음 `create-resource-share` 예제에서는 지정된 이름으로 빈 리소스 공유를 생성합니다. 공유에 리소스, 보안 주체 및 권한을 별도로 추가해야 합니다.

```
aws ram create-resource-share \
    --name MyNewResourceShare
```

출력:

```
{
    "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
```

```
    }
}
```

예제 2: AWS 계정을 보안 주체로 사용하여 리소스 공유를 생성하려면

다음 create-resource-share 예제에서는 리소스 공유를 생성하고 지정된 AWS 계정(222222222222)에 대한 액세스 권한을 부여합니다. 지정된 보안 주체가 동일한 AWS 조직의 일부가 아닌 경우 초대가 전송되며 액세스 권한이 부여되기 전에 수락되어야 합니다.

```
aws ram create-resource-share \
    --name MyNewResourceShare \
    --principals 222222222222
```

예제 3: AWS 조직으로 제한된 리소스 공유 생성

다음 create-resource-share 예제에서는 계정이 구성원인 AWS 조직의 계정으로 제한된 리소스 공유를 생성하고 지정된 OU를 보안 주체로 추가합니다. 해당 OU의 모든 계정은 리소스 공유의 리소스를 사용할 수 있습니다.

```
aws ram create-resource-share \
    --name MyNewResourceShare \
    --no-allow-external-principals \
    --principals arn:aws:organizations::123456789012:ou/o-63bEXAMPLE/ou-46xi-
rEXAMPLE
```

출력:

```
{
    "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
    }
}
```

- 자세한 API 내용은 명령 참조CreateResourceShare의 섹션을 참조하세요. AWS CLI

### delete-resource-share

다음 코드 예시에서는 delete-resource-share을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 공유를 삭제하려면

다음 delete-resource-share 예제에서는 지정된 리소스 공유를 삭제합니다.

```
aws ram delete-resource-share \
    --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE
```

다음 출력은 성공을 나타냅니다.

```
{
    "returnValue": true
}
```

- 자세한 API 내용은 명령 참조DeleteResourceShare의 섹션을 참조하세요. AWS CLI

### disassociate-resource-share-permission

다음 코드 예시에서는 disassociate-resource-share-permission을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 공유에서 리소스 유형에 대한 RAM 관리형 권한을 제거하려면

다음 disassociate-resource-share-permission 예제에서는 지정된 리소스 공유에서 Glue 데이터베이스에 대한 RAM 관리형 권한을 제거합니다.

```
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
```

출력:

```
{
    "returnValue": true
}
```

- 자세한 API 내용은 명령 참조DisassociateResourceSharePermission의 섹션을 참조하세요. AWS CLI

### disassociate-resource-share

다음 코드 예시에서는 disassociate-resource-share을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 공유에서 리소스를 제거하려면

다음 disassociate-resource-share 예제에서는 지정된 리소스 공유에서 지정된 리소스, 이 경우 VPC 서브넷을 제거합니다. 리소스 공유에 액세스할 수 있는 보안 주체는 더 이상 해당 리소스에 대한 작업을 수행할 수 없습니다.

```
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
```

출력:

```
{
    "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
    ]
}
```

- 자세한 API 내용은 명령 참조DisassociateResourceShare의 섹션을 참조하세요. AWS CLI

### enable-sharing-with-aws-organization

다음 코드 예시에서는 enable-sharing-with-aws-organization을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 조직 간에 리소스 공유를 활성화하려면

다음 enable-sharing-with-aws-organization 예제에서는 조직 및 조직 단위 간에 리소스 공유를 활성화합니다.

```
aws ram enable-sharing-with-aws-organization
```

다음 출력은 성공을 나타냅니다.

```
{
    "returnValue": true
}
```

- 자세한 API 내용은 명령 참조EnableSharingWithAwsOrganization의 섹션을 참조하세요. AWS CLI

### get-permission

다음 코드 예시에서는 get-permission을 사용하는 방법을 보여 줍니다.

AWS CLI

RAM 관리형 권한에 대한 세부 정보를 검색하려면

다음 get-permission 예제에서는 지정된 RAM 관리형 권한의 기본 버전에 대한 세부 정보를 표시합니다.

```
aws ram get-permission \
    --permission-arn arn:aws:ram::aws:permission/
AWSRAMPermissionGlueTableReadWriteForDatabase
```

출력:

```
{
    "permission": {
        "arn": "arn:aws:ram::aws:permission/
AWSRAMPermissionGlueTableReadWriteForDatabase",
        "version": "2",
        "defaultVersion": true,
        "name": "AWSRAMPermissionGlueTableReadWriteForDatabase",
        "resourceType": "glue:Database",
        "permission": "{\"Effect\":\"Allow\",\"Action\":[\"glue:GetTable
\", \"glue:UpdateTable\", \"glue:DeleteTable\", \"glue:BatchDeleteTable\",
 \"glue:BatchDeleteTableVersion\", \"glue:GetTableVersion\", \"glue:GetTableVersions
\", \"glue:GetPartition\", \"glue:GetPartitions\", \"glue:BatchGetPartition\",
 \"glue:BatchCreatePartition\", \"glue:CreatePartition\", \"glue:UpdatePartition
\", \"glue:BatchDeletePartition\", \"glue:DeletePartition\", \"glue:GetTables\",
 \"glue:SearchTables\"]}",
        "creationTime": 1624912434.431,
        "lastUpdatedTime": 1624912434.431,
        "isResourceTypeDefault": false
    }
}
```

- 자세한 API 내용은 명령 참조GetPermission의 섹션을 참조하세요. AWS CLI

### get-resource-policies

다음 코드 예시에서는 get-resource-policies을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 대한 정책을 가져오려면

다음 get-resource-policies 예제에서는 리소스 공유와 연결된 지정된 리소스에 대한 리소스 기반 권한 정책을 보여줍니다.

```
aws ram get-resource-policies \
    --resource-arns arn:aws:ec2:us-west-2:123456789012:subnet/
subnet-0250c25a1fEXAMPLE
```

출력:

```
{
    "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\"}]}"
    ]
}
```

- 자세한 API 내용은 명령 참조GetResourcePolicies의 섹션을 참조하세요. AWS CLI

### get-resource-share-associations

다음 코드 예시에서는 get-resource-share-associations을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 모든 리소스 유형에 대한 모든 리소스 연결을 나열하려면

다음 get-resource-share-associations 예제에서는 모든 리소스 공유의 모든 리소스 유형에 대한 리소스 연결을 나열합니다.

```
aws ram get-resource-share-associations \
    --association-type RESOURCE
```

출력:

```
{
    "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
        }
    ]
}
```

예제 2: 리소스 공유의 보안 주체 연결을 나열하려면

다음 `get-resource-share-associations` 예제에서는 지정된 리소스 공유에 대한 보안 주체 연결만 나열합니다.

```
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
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetResourceShareAssociations의 섹션을 참조하세요. AWS CLI

### get-resource-share-invitations

다음 코드 예시에서는 get-resource-share-invitations을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 공유 초대를 나열하려면

다음 get-resource-share-invitations 예제에서는 현재 리소스 공유 초대를 나열합니다.

```
aws ram get-resource-share-invitations
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetResourceShareInvitations의 섹션을 참조하세요. AWS CLI

### get-resource-shares

다음 코드 예시에서는 get-resource-shares을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 소유한 리소스 공유를 나열하고 다른 사용자와 공유하려면

다음 get-resource-shares 예제에서는 다른 사용자와 공유하고 생성한 리소스 공유를 나열합니다.

```
aws ram get-resource-shares \
    --resource-owner SELF
```

출력:

```
{
    "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
        }
    ]
}
```

예제 2: 다른 사용자가 소유하고 사용자와 공유한 리소스 공유를 나열하려면

다음 `get-resource-shares` 예제에서는 다른 사용자가 생성하고 공유한 리소스 공유를 나열합니다. 이 예제에서는 아무 것도 없습니다.

```
aws ram get-resource-shares \
```

```
    --resource-owner OTHER-ACCOUNTS
```

출력:

```
{
    "resourceShares": []
}
```

- 자세한 API 내용은 명령 참조GetResourceShares의 섹션을 참조하세요. AWS CLI

### `list-pending-invitation-resources`

다음 코드 예시에서는 `list-pending-invitation-resources`을 사용하는 방법을 보여 줍니다.

AWS CLI

보류 중인 리소스 공유에서 사용할 수 있는 리소스를 나열하려면

다음 `list-pending-invitation-resources` 예제에서는 지정된 초대와 연결된 리소스 공유에 있는 모든 리소스를 나열합니다.

```
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
```

출력:

```
{
   "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListPendingInvitationResources의 섹션을 참조하세요. AWS CLI

### `list-permissions`

다음 코드 예시에서는 `list-permissions`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 RAM 관리형 권한을 나열하려면

다음 `list-permissions` 예제에서는 AWS Glue 데이터베이스 리소스 유형에만 사용할 수 있는 모든 RAM 관리형 권한을 나열합니다.

```
aws ram list-permissions \
    --resource-type glue:Database
```

출력:

```
{
    "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
        }
    ]
}
```

다음 `list-permissions` 예제에서는 모든 리소스 유형에 대해 사용 가능한 RAM 관리형 권한을 표시합니다.

```
aws ram list-permissions
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListPermissions의 섹션을 참조하세요. AWS CLI

### list-principals

다음 코드 예시에서는 list-principals을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 액세스할 수 있는 보안 주체를 나열하려면

다음 list-principals 예제에서는 모든 리소스 공유를 통해 지정된 유형의 리소스에 액세스할 수 있는 보안 주체 목록을 표시합니다.

```
aws ram list-principals \
    --resource-type ec2:Subnet
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListPrincipals의 섹션을 참조하세요. AWS CLI

### list-resource-share-permissions

다음 코드 예시에서는 list-resource-share-permissions을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 공유에 현재 연결된 모든 RAM 관리형 권한을 나열하려면

다음 list-resource-share-permissions 예제에서는 지정된 리소스 공유에 연결된 모든 RAM 관리형 권한을 나열합니다.

```
aws ram list-resource-share-permissions \
    --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-
share/27d09b4b-5e12-41d1-a4f2-19dedEXAMPLE
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListResourceSharePermissions의 섹션을 참조하세요. AWS CLI

### `list-resource-types`

다음 코드 예시에서는 `list-resource-types`을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 지원하는 리소스 유형을 나열하려면 AWS RAM

다음 `list-resource-types` 예제에서는 현재 에서 지원하는 모든 리소스 유형을 나열합니다 AWS RAM.

```
aws ram list-resource-types
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListResourceTypes의 섹션을 참조하세요. AWS CLI

### `list-resources`

다음 코드 예시에서는 `list-resources`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 공유와 연결된 리소스를 나열하려면

다음 `list-resources` 예제에서는 지정된 리소스 공유에 있는 지정된 리소스 유형의 모든 리소스를 나열합니다.

```
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
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListResources의 섹션을 참조하세요. AWS CLI

### promote-resource-share-created-from-policy

다음 코드 예시에서는 promote-resource-share-created-from-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 정책 기반 리소스 공유를 의 전체 기능으로 승격하려면 AWS RAM

다음 promote-resource-share-created-from-policy 예제에서는 리소스 기반 정책을 연결하여 암시적으로 생성한 리소스 공유를 가져와서 콘솔 및 해당 CLI 및 API 작업에서 AWS RAM 완전히 작동하도록 변환합니다.

```
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
```

출력:

```
{
    "returnValue": true
}
```

- 자세한 API 내용은 명령 참조PromoteResourceShareCreatedFromPolicy의 섹션을 참조하세요. AWS CLI

### `reject-resource-share-invitation`

다음 코드 예시에서는 `reject-resource-share-invitation`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 공유 초대를 거부하려면

다음 `reject-resource-share-invitation` 예제에서는 지정된 리소스 공유 초대를 거부합니다.

```
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
```

출력:

```
"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"
    }
]
```

- 자세한 API 내용은 명령 참조RejectResourceShareInvitation의 섹션을 참조하세요. AWS CLI

### `tag-resource`

다음 코드 예시에서는 `tag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 공유에 태그를 추가하려면

다음 tag-resource 예제에서는 태그 키project와 관련 값을 지정된 리소스 공유lima에 추가합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### untag-resource

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 공유에서 태그를 제거하려면

다음 untag-resource 예제에서는 지정된 리소스 공유에서 project 태그 키와 관련 값을 제거합니다.

```
aws ram untag-resource \
    --tag-keys project \
    --resource-share-arn arn:aws:ram:us-west-2:123456789012:resource-share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-resource-share

다음 코드 예시에서는 update-resource-share을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 공유를 업데이트하려면

다음 update-resource-share 예제에서는 지정된 리소스 공유를 변경하여 AWS 조직에 없는 외부 보안 주체를 허용합니다.

```
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
```

출력:

```
{
    "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
    }
}
```

- 자세한 API 내용은 명령 참조UpdateResourceShare의 섹션을 참조하세요. AWS CLI

## 를 사용한 Resource Explorer 예제 AWS CLI

다음 코드 예제에서는 Resource Explorer AWS Command Line Interface 에서 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### `associate-default-view`

다음 코드 예시에서는 `associate-default-view`을 사용하는 방법을 보여 줍니다.

AWS CLI

Resource Explorer 뷰를 해당 AWS 리전의 기본값으로 설정하려면

다음 `associate-default-view` 예제에서는 에 지정된 뷰를 작업을 호출하는 AWS 리전의 ARN 기본 뷰로 설정합니다.

```
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
```

출력:

```
{
    "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-Main-
View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"
}
```

자세한 내용은 AWS Resource Explorer 사용 설명서의 AWS 리전에서 기본 보기 설정을 참조하세요.

- 자세한 API 내용은 명령 참조AssociateDefaultView의 섹션을 참조하세요. AWS CLI

### `batch-get-view`

다음 코드 예시에서는 `batch-get-view`을 사용하는 방법을 보여 줍니다.

AWS CLI

여러 Resource Explorer 뷰에 대한 세부 정보를 검색하려면

다음 `batch-get-view` 예제에서는 에서 지정한 두 뷰에 대한 세부 정보를 표시합니다ARNs. 공백을 사용하여 --view-arn 파라미터ARNs의 다중 를 구분합니다.

```
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
```

출력:

```
{
    "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": []
}
```

뷰에 대한 자세한 내용은 Resource Explorer 사용 설명서의 Resource Explorer 뷰 정보를 참조하세요. AWS

- 자세한 API 내용은 명령 참조BatchGetView의 섹션을 참조하세요. AWS CLI

### `create-index`

다음 코드 예시에서는 `create-index`을 사용하는 방법을 보여 줍니다.

AWS CLI

인덱스를 생성하여 AWS 리전에서 Resource Explorer를 켜려면

다음 `create-index` 예제에서는 작업이 호출되는 AWS 리전에 로컬 인덱스를 생성합니다. 는 AWS CLI 무작위 `client-token` 파라미터 값을 자동으로 생성하고 값을 지정 AWS 하지 않으면 에 대한 호출에 포함합니다.

```
aws resource-explorer-2 create-index \
    --region us-east-1
```

출력:

```
{
    "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"
}
```

로컬 인덱스를 생성한 후 update-index-type 명령을 실행하여 계정의 애그리게이터 인덱스로 변환할 수 있습니다.

자세한 내용은 Resource Explorer 사용 설명서의 리소스 인덱싱을 위해 AWS 리전에서 Resource Explorer 켜기를 참조하세요. AWS

- 자세한 API 내용은 명령 참조CreateIndex의 섹션을 참조하세요. AWS CLI

### `create-view`

다음 코드 예시에서는 `create-view`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: AWS 리전의 인덱스에 대해 필터링되지 않은 뷰를 생성하려면

다음 create-view 예제에서는 지정된 AWS 리전에 필터링 없이 리전의 모든 결과를 반환하는 뷰를 생성합니다. 뷰에는 반환된 결과에 대한 선택적 태그 필드가 포함됩니다. 이 보기는 집계자 인덱스가 포함된 리전에서 생성되므로 Resource Explorer 인덱스가 포함된 계정의 모든 리전의 결과를 포함할 수 있습니다.

```
aws resource-explorer-2 create-view \
    --view-name My-Main-View \
    --included-properties Name=tags \
    --region us-east-1
```

출력:

```
{
    "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"
    }
}
```

예제 2: Amazon과 연결된 리소스만 반환하는 보기를 생성하려면 EC2

다음은 Amazon EC2 서비스와 연결된 AWS 리전의 리소스만 us-east-1 반환하는 보기를 리전에 create-view 생성합니다. 뷰에는 반환된 결과에 대한 선택적 Tags 필드가 포함됩니다. 이 보기는 집계자 인덱스가 포함된 리전에서 생성되므로 Resource Explorer 인덱스가 포함된 계정의 모든 리전의 결과를 포함할 수 있습니다.

```
aws resource-explorer-2 create-view \
    --view-name My-EC2-Only-View \
    --included-properties Name=tags \
    --filters FilterString="service:ec2" \
```

```
    --region us-east-1
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS Resource Explorer 사용 설명서의 검색을 위한 뷰 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateView의 섹션을 참조하세요. AWS CLI

### delete-index

다음 코드 예시에서는 delete-index을 사용하는 방법을 보여 줍니다.

AWS CLI

인덱스를 삭제하여 AWS 리전에서 Resource Explorer를 끄려면

다음 delete-index 예제에서는 요청을 수행하는 AWS 리전에서 지정된 Resource Explorer 인덱스를 삭제합니다.

```
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
```

출력:

```
{
    "Arn": "arn:aws:resource-explorer-2:us-west-2:123456789012:index/EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222",
    "State": "DELETING"
}
```

인덱스 삭제에 대한 자세한 내용은 AWS Resource Explorer 사용 설명서의 AWS 리전에서 Resource Explorer 끄기를 참조하세요. AWS

- 자세한 API 내용은 명령 참조DeleteIndex의 섹션을 참조하세요. AWS CLI

### `delete-view`

다음 코드 예시에서는 `delete-view`을 사용하는 방법을 보여 줍니다.

AWS CLI

Resource Explorer 보기를 삭제하려면

다음 `delete-view` 예제에서는 에서 지정한 보기를 삭제합니다ARN.

```
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
```

출력:

```
{
    "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/EC2-Only-View/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"
}
```

자세한 내용은 AWS Resource Explorer 사용 설명서의 보기 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteView의 섹션을 참조하세요. AWS CLI

### `disassociate-default-view`

다음 코드 예시에서는 `disassociate-default-view`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 리전에 대한 기본 Resource Explorer 보기를 제거하려면

다음은 작업을 호출하는 AWS 리전의 기본 Resource Explorer 보기를 `disassociate-default-view` 제거합니다. 이 작업을 수행한 후 리전의 모든 검색 작업은 뷰를 명시적으로 지정해야 합니다. 그렇지 않으면 작업이 실패합니다.

```
aws resource-explorer-2 disassociate-default-view
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Resource Explorer 사용 설명서의 AWS 리전에서 기본 보기 설정을 참조하세요.

- 자세한 API 내용은 명령 참조DisassociateDefaultView의 섹션을 참조하세요. AWS CLI

### `get-default-view`

다음 코드 예시에서는 `get-default-view`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 해당 리전의 기본 뷰인 Resource Explorer 뷰를 검색하려면

다음 `get-default-view` 예제에서는 작업을 호출하는 AWS 리전의 기본값인 뷰ARN의 를 검색합니다.

```
aws resource-explorer-2 get-default-view
```

출력:

```
{
    "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/default-view/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"
}
```

자세한 내용은 AWS Resource Explorer 사용 설명서의 AWS 리전에서 기본 보기 설정을 참조하세요.

- 자세한 API 내용은 명령 참조GetDefaultView의 섹션을 참조하세요. AWS CLI

### get-index

다음 코드 예시에서는 get-index을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: Resource Explorer 애그리게이터 인덱스의 세부 정보를 검색하려면

다음 get-index 예제에서는 지정된 AWS 리전의 Resource Explorer 인덱스에 대한 세부 정보를 표시합니다. 지정된 리전에는 계정의 집계기 인덱스가 포함되어 있으므로 출력에는 이 리전의 인덱스에 데이터를 복제하는 리전이 나열됩니다.

```
aws resource-explorer-2 get-index \
    --region us-east-1
```

출력:

```
{
    "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"
}
```

예제 2: Resource Explorer 로컬 인덱스에 대한 세부 정보를 검색하려면

다음 get-index 예제에서는 지정된 AWS 리전의 Resource Explorer 인덱스에 대한 세부 정보를 표시합니다. 지정된 리전에는 로컬 인덱스가 포함되어 있으므로 출력에는 이 리전의 인덱스에서 데이터를 복제하는 리전이 나열됩니다.

```
aws resource-explorer-2 get-index \
```

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

출력:

```
{
    "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"
}
```

인덱스에 대한 자세한 내용은 Resource Explorer 사용 설명서의 Resource Explorer가 켜져 있는 AWS 리전 확인을 참조하세요. AWS

- 자세한 API 내용은 명령 참조GetIndex의 섹션을 참조하세요. AWS CLI

### get-view

다음 코드 예시에서는 get-view을 사용하는 방법을 보여 줍니다.

AWS CLI

Resource Explorer 뷰에 대한 세부 정보를 검색하려면

다음 get-view 예제에서는 에서 지정한 뷰에 대한 세부 정보를 표시합니다ARN.

```
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
```

출력:

```
{
  "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"
  }
}
```

뷰에 대한 자세한 내용은 Resource Explorer 사용 설명서의 Resource Explorer 뷰 정보를 참조하세요. AWS

- 자세한 API 내용은 명령 참조GetView의 섹션을 참조하세요. AWS CLI

### `list-indexes`

다음 코드 예시에서는 `list-indexes`을 사용하는 방법을 보여 줍니다.

AWS CLI

Resource Explorer에 인덱스가 있는 AWS 리전을 나열하려면

다음 `list-indexes` 예제에서는 Resource Explorer에 인덱스가 있는 모든 리전의 인덱스를 나열합니다. 응답은 각 인덱스의 유형, AWS 리전 및 를 지정합니다ARN.

```
aws resource-explorer-2 list-indexes
```

출력:

```
{
    "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"
        }
    ]
}
```

인덱스에 대한 자세한 내용은 Resource Explorer 사용 설명서의 Resource Explorer가 켜져 있는 AWS 리전 확인을 참조하세요. AWS

- 자세한 API 내용은 명령 참조ListIndexes의 섹션을 참조하세요. AWS CLI

### `list-supported-resource-types`

다음 코드 예시에서는 `list-supported-resource-types`을 사용하는 방법을 보여 줍니다.

AWS CLI

Resource Explorer에 인덱스가 있는 AWS 리전을 나열하려면

다음 `list-supported-resource-types` 예제에서는 현재 &AREXlong;에서 지원하는 모든 리소스 유형을 나열합니다. 예제 응답에는 추가 호출로 검색할 수 있는 출력이 더 많음을 나타내는 `NextToken` 값이 포함됩니다.

```
aws resource-explorer-2 list-supported-resource-types \
    --max-items 10
```

출력:

```
{
    "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": "eyJOZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxMH0="
}
```

출력의 다음 부분을 가져오려면 작업을 다시 호출하고 이전 호출의 NextToken 응답 값을 의 값으로 전달합니다--starting-token. NextToken 가 응답에 없을 때까지 반복합니다.

```
aws resource-explorer-2 list-supported-resource-types \
    --max-items 10  \
    --starting-
token eyJOZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxMH0=
```

출력:

```
{
    "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": "eyJOZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyMH0="
}
```

인덱스에 대한 자세한 내용은 Resource Explorer 사용 설명서의 Resource Explorer가 켜져 있는 AWS 리전 확인을 참조하세요. AWS

- 자세한 API 내용은 명령 참조ListSupportedResourceTypes의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

Resource Explorer 뷰 또는 인덱스에 연결된 태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 지정된 로 보기 위해 연결된 태그 키와 값 페어를 나열합니다ARN. 리소스가 포함된 AWS 리전에서 작업을 호출해야 합니다.

```
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
```

출력:

```
{
    "Tags": {
        "application": "MainCorpApp",
```

```
        "department": "1234"
    }
}
```

뷰 태그 지정에 대한 자세한 내용은 AWS Resource Explorer 사용 설명서의 액세스 제어를 위한 뷰 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### `list-views`

다음 코드 예시에서는 `list-views`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 리전에서 사용할 수 있는 Resource Explorer 뷰를 나열하려면

다음 `list-views` 예제에서는 작업을 호출하는 리전에서 사용할 수 있는 모든 뷰를 나열합니다.

```
aws resource-explorer-2 list-views
```

출력:

```
{
    "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"
    ]
}
```

뷰에 대한 자세한 내용은 Resource Explorer 사용 설명서의 Resource Explorer 뷰 정보를 참조하세요. AWS

- 자세한 API 내용은 명령 참조ListViews의 섹션을 참조하세요. AWS CLI

### `search`

다음 코드 예시에서는 `search`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 기본 보기를 사용하여 검색하려면

다음 search 예제에서는 서비스와 연결된 지정된 의 모든 리소스를 표시합니다. 검색은 리전에 대한 기본 보기를 사용합니다. 예제 응답에는 추가 호출로 검색할 수 있는 출력이 더 많음을 나타내는 NextToken 값이 포함됩니다.

```
aws resource-explorer-2 search \
    --query-string "service:iam"
```

출력:

```
{
    "Count": {
        "Complete": true,
        "TotalResources": 55
    },
    "NextToken":
 "AG9VOEF1KLEXAMPLEOhJHVwo5chEXAMPLER5XiEpNrgsEXAMPLE...b0CmOFOryHEXAMPLE",
    "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"
```

```
}
```

예제 2: 지정된 보기를 사용하여 검색하려면

다음 search 예제 검색은 지정된 뷰를 통해 표시되는 지정된 AWS 리전의 모든 리소스(“*”)를 표시합니다. 결과에는 뷰에 연결된 필터로 EC2 인해 Amazon과 연결된 리소스만 포함됩니다.

```
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
```

출력:

```
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:dhcpoptions",
            "Service": "ec2"
        }, {
            ... TRUNCATED FOR BREVITY ...
        }],
        "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-EC2-
view/EXAMPLE8-90ab-cdef-fedc-EXAMPLE22222"
    }
```

자세한 내용은 AWS Resource Explorer 사용 설명서의 Resource Explorer를 사용하여 리소스를 검색하기를 참조하세요. AWS

- API 자세한 내용은 명령 참조의 검색을 참조하세요. AWS CLI

### `tag-resource`

다음 코드 예시에서는 `tag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

Resource Explorer 뷰에 태그를 지정하려면

다음 `tag-resource` 예제에서는 태그 키 '환경'과 '프로덕션' 값을 지정된 가 있는 뷰에 추가합니다 ARN.

```
aws resource-explorer-2 tag-resource \
    --resource-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-View//
EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111 \
    --tags environment=production
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Resource Explorer 사용 설명서의 액세스 제어를 위한 보기 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### `untag-resource`

다음 코드 예시에서는 `untag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

Resource Explorer 보기에서 태그를 제거하려면

다음 `untag-resource` 예제에서는 키 이름이 '환경'인 태그를 지정된 가 있는 뷰에서 제거합니다 ARN.

```
aws resource-explorer-2 untag-resource \
    --resource-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-View//
EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111 \
```

```
    --tag-keys environment
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Resource Explorer 사용 설명서의 액세스 제어를 위한 뷰 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-index-type

다음 코드 예시에서는 update-index-type을 사용하는 방법을 보여 줍니다.

AWS CLI

Resource Explorer 인덱스의 유형을 변경하려면

다음 update-index-type 예제에서는 지정된 인덱스를 유형에서 유형local으로 변환aggregator하여 계정의 모든 AWS 리전에서 리소스를 검색할 수 있는 기능을 활성화합니다. 업데이트하려는 인덱스가 포함된 AWS 리전으로 요청을 보내야 합니다.

```
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
```

출력:

```
{
    "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"
}
```

인덱스 유형 변경에 대한 자세한 내용은 AWS Resource Explorer 사용 설명서의 집계기 인덱스를 생성하여 리전 간 검색 켜기를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateIndexType의 섹션을 참조하세요. AWS CLI

**update-view**

다음 코드 예시에서는 update-view을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: Resource Explorer 보기의 IncludedProperties 필드를 업데이트하려면

다음 update-view 예제에서는 선택적 에 를 추가하여 지정된 뷰를 업데이트`tags`합니다`IncludedProperties`. 이 작업을 실행한 후 이 보기를 사용하는 검색 작업에는 결과에 표시되는 리소스에 연결된 태그에 대한 정보가 포함됩니다.

```
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
```

출력:

```
{
    "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"
    }
}
```

예제 2: 뷰에 연결된 필터를 업데이트하려면

다음 update-view 예제에서는 결과를 Amazon EC2 서비스와 연결된 리소스 유형으로만 제한하는 필터를 사용하도록 지정된 보기를 업데이트합니다.

```
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
```

출력:

```
{
    "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"
    }
}
```

뷰에 대한 자세한 내용은 Resource Explorer 사용 설명서의 Resource Explorer 뷰 정보를 참조하세요. AWS

- 자세한 API 내용은 명령 참조UpdateView의 섹션을 참조하세요. AWS CLI

## 를 사용한 리소스 그룹 예제 AWS CLI

다음 코드 예제에서는 리소스 그룹과 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### `create-group`

다음 코드 예시에서는 create-group을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 태그 기반 리소스 그룹을 생성하려면

다음 create-group 예제에서는 현재 리전에 Amazon EC2 인스턴스의 태그 기반 리소스 그룹을 생성합니다. 키 Name, 값 로 태그가 지정된 리소스에 대한 쿼리를 기반으로 합니다WebServers. 그룹 이름은 입니다tbq-WebServer. 쿼리는 명령에 전달되는 별도의 JSON 파일에 있습니다.

```
aws resource-groups create-group \
    --name tbq-WebServer \
    --resource-query file://query.json
```

query.json의 콘텐츠:

```
{
    "Type": "TAG_FILTERS_1_0",
    "Query": "{\"ResourceTypeFilters\":[\"AWS::EC2::Instance\"],\"TagFilters\":
[{\"Key\":\"Name\", \"Values\":[\"WebServers\"]}]}"
}
```

출력:

```
{
    "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\"]}]}"
    }
}
```

예제 2: CloudFormation 스택 기반 리소스 그룹을 생성하려면

다음 create-group 예제에서는 라는 AWS CloudFormation 스택 기반 리소스 그룹을 생성합니다sampleCFNstackgroup. 쿼리에는 리소스 그룹에서 지원하는 지정된 CloudFormation 스택의 모든 AWS 리소스가 포함됩니다.

```
aws resource-groups create-group \
    --name cbq-CFNstackgroup \
    --resource-query file://query.json
```

query.json의 콘텐츠:

```
{
    "Type": "CLOUDFORMATION_STACK_1_0",
    "Query": "{\"ResourceTypeFilters\":[\"AWS::AllSupported\"],\"StackIdentifier
\":\"arn:aws:cloudformation:us-west-2:123456789012:stack/MyCFNStack/1415z9z0-
z39z-11z8-97z5-500z212zz6fz\"}"
}
```

출력:

```
{
    "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\"}"}'
    }
}
```

자세한 내용은 리소스 그룹 사용 설명서의 그룹 생성을 참조하세요. AWS

- 자세한 API 내용은 명령 참조CreateGroup의 섹션을 참조하세요. AWS CLI

### delete-group

다음 코드 예시에서는 delete-group을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 그룹에 대한 설명을 업데이트하려면

다음 delete-group 예제에서는 지정된 리소스 그룹을 업데이트합니다.

```
aws resource-groups delete-group \
    --group-name tbq-WebServer
```

출력:

```
{
    "Group": {
        "GroupArn": "arn:aws:resource-groups:us-west-2:1234567890:group/tbq-
WebServer",
        "Name": "tbq-WebServer"
    }
}
```

자세한 내용은 리소스 그룹 사용 설명서의 그룹 삭제를 참조하세요. AWS

- 자세한 API 내용은 명령 참조DeleteGroup의 섹션을 참조하세요. AWS CLI

### get-group-query

다음 코드 예시에서는 get-group-query을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 그룹에 쿼리를 연결하려면

다음 get-group-query 예제에서는 지정된 리소스 그룹에 연결된 쿼리를 표시합니다.

```
aws resource-groups get-group-query \
    --group-name tbq-WebServer
```

출력:

```
{
    "GroupQuery": {
        "GroupName": "tbq-WebServer",
```

```
        "ResourceQuery": {
            "Type": "TAG_FILTERS_1_0",
            "Query": "{\"ResourceTypeFilters\":[\"AWS::EC2::Instance\"],\"TagFilters
\":[{\"Key\":\"Name\", \"Values\":[\"WebServers\"]}]}"
        }
    }
}
```

- 자세한 API 내용은 명령 참조GetGroupQuery의 섹션을 참조하세요. AWS CLI

### get-group

다음 코드 예시에서는 get-group을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 그룹에 대한 정보를 가져오려면

다음 get-group 예제에서는 지정된 리소스 그룹에 대한 세부 정보를 표시합니다. 쿼리를 그룹에 연결하려면 를 사용합니다get-group-query.

```
aws resource-groups get-group \
    --group-name tbq-WebServer
```

출력:

```
{
    "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."
    }
}
```

- 자세한 API 내용은 명령 참조GetGroup의 섹션을 참조하세요. AWS CLI

### get-tags

다음 코드 예시에서는 get-tags을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 그룹에 연결된 태그를 검색하려면

다음 get-tags 예제에서는 지정된 리소스 그룹(그룹 자체, 구성원 아님)에 연결된 태그 키와 값 페어를 표시합니다.

```
aws resource-groups get-tags \
    --arn arn:aws:resource-groups:us-west-2:123456789012:group/tbq-WebServer
```

출력:

```
{
    "Arn": "arn:aws:resource-groups:us-west-2:123456789012:group/tbq-WebServer",
    "Tags": {
        "QueryType": "tags",
        "QueryResources": "ec2-instances"
    }
}
```

- 자세한 API 내용은 명령 참조GetTags의 섹션을 참조하세요. AWS CLI

### list-group-resources

다음 코드 예시에서는 list-group-resources을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 그룹의 모든 리소스를 나열하려면

예제 1: 다음 list-resource-groups 예제에서는 지정된 리소스 그룹의 일부인 모든 리소스를 나열합니다.

```
aws resource-groups list-group-resources \
    --group-name tbq-WebServer
```

출력:

```
{
    "ResourceIdentifiers": [
        {
```

```
            "ResourceArn": "arn:aws:ec2:us-west-2:123456789012:instance/
i-09f77fa38c12345ab",
            "ResourceType": "AWS::EC2::Instance"
        }
    ]
}
```

예제 2: 다음 예제에서는 ':AWS::EC2Instance'의 '리소스 유형'도 있는 그룹의 모든 리소스를 나열합니다. :

aws resource-groups list-group-resources --group-name tbq-WebServer -filters Name=resource-type,Values=AWS::EC2:Instance

- 자세한 API 내용은 명령 참조ListGroupResources의 섹션을 참조하세요. AWS CLI

### `list-groups`

다음 코드 예시에서는 `list-groups`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 리소스 그룹을 나열하려면

다음 `list-groups` 예제에서는 모든 리소스 그룹의 목록을 표시합니다.

```
aws resource-groups list-groups
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListGroups의 섹션을 참조하세요. AWS CLI

### `list-resource-groups`

다음 코드 예시에서는 `list-resource-groups`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 그룹의 모든 리소스를 나열하려면

다음 `list-resource-groups` 예제에서는 지정된 리소스 그룹의 일부인 모든 리소스를 나열합니다.

```
aws resource-groups list-group-resources \
    --group-name tbq-WebServer
```

출력:

```
{
    "ResourceIdentifiers": [
        {
            "ResourceArn": "arn:aws:ec2:us-west-2:123456789012:instance/
i-09f77fa38c12345ab",
            "ResourceType": "AWS::EC2::Instance"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListResourceGroups의 섹션을 참조하세요. AWS CLI

### put-group-configuration

다음 코드 예시에서는 put-group-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 그룹에 서비스 구성을 연결하려면

예제 1: 다음 put-group-configuration 예제에서는 리소스 그룹이 C5 또는 M5 패밀리의 인스턴스에 대한 Amazon EC2 용량 예약만 포함하도록 지정합니다.

```
aws resource-groups put-group-configuration \
    --group MyTestGroup \
    --configuration file://config.json
```

config.json의 콘텐츠:

```
[
    {
        "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" ]
            }
        ]
    }
]
```

이 명령은 성공 시 출력을 생성하지 않습니다.

자세한 내용은 리소스 그룹 참조 가이드의 리소스 그룹에 대한 서비스 구성을 참조하세요API.

- 자세한 API 내용은 명령 참조PutGroupConfiguration의 섹션을 참조하세요. AWS CLI

### `search-resources`

다음 코드 예시에서는 `search-resources`을 사용하는 방법을 보여 줍니다.

AWS CLI

쿼리와 일치하는 리소스를 찾으려면

다음 `search-resources` 예제에서는 지정된 쿼리와 일치하는 모든 AWS 리소스 목록을 검색합니다.

```
aws resource-groups search-resources \
    --resource-query file://query.json
```

`query.json`의 콘텐츠:

```
{
    "Type": "TAG_FILTERS_1_0",
    "Query": "{\"ResourceTypeFilters\":[\"AWS::EC2::Instance\"],\"TagFilters\":
[{\"Key\":\"Patch Group\", \"Values\":[\"Dev\"]}]}"
}
```

출력:

```
{
    "ResourceIdentifiers": [
        {
            "ResourceArn": "arn:aws:ec2:us-west-2:123456789012:instance/
i-01a23bc45d67890ef",
            "ResourceType": "AWS::EC2::Instance"
```

```
        }
    ]
}
```

- 자세한 API 내용은 명령 참조SearchResources의 섹션을 참조하세요. AWS CLI

### `tag`

다음 코드 예시에서는 tag을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 그룹에 태그를 연결하려면

다음 tag 예제에서는 지정된 태그 키와 값 페어를 지정된 리소스 그룹(그룹 자체, 구성원 아님)에 연결합니다.

```
aws resource-groups tag \
    --tags QueryType=tags,QueryResources=ec2-instances \
    --arn arn:aws:resource-groups:us-west-2:128716708097:group/tbq-WebServer
```

출력:

```
{
    "Arn": "arn:aws:resource-groups:us-west-2:128716708097:group/tbq-WebServer",
    "Tags": {
        "QueryType": "tags",
        "QueryResources": "ec2-instances"
    }
}
```

자세한 내용은 리소스 그룹 사용 설명서의 태그 관리를 참조하세요. AWS

- API 자세한 내용은 명령 참조의 태그를 참조하세요. AWS CLI

### `untag`

다음 코드 예시에서는 untag을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 그룹에서 태그를 제거하려면

다음 untags 예제에서는 지정된 키가 있는 태그를 리소스 그룹 자체에서 제거합니다.

```
aws resource-groups untag \
    --arn arn:aws:resource-groups:us-west-2:123456789012:group/tbq-WebServer \
    --keys QueryType
```

출력:

```
{
    "Arn": "arn:aws:resource-groups:us-west-2:123456789012:group/tbq-WebServer",
    "Keys": [
        "QueryType"
    ]
}
```

자세한 내용은 리소스 그룹 사용 설명서의 태그 관리를 참조하세요. AWS

- API 자세한 내용은 AWS CLI 명령 참조의 태그 해제를 참조하세요.

### update-group-query

다음 코드 예시에서는 update-group-query을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 태그 기반 리소스 그룹에 대한 쿼리를 업데이트하려면

다음 update-group-query 예제에서는 지정된 태그 기반 리소스 그룹에 연결된 쿼리를 업데이트합니다.

```
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
\"]}]}"}'
```

출력:

```
{
    "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\"]}]}"
    }
}
```

자세한 내용은 리소스 그룹 사용 설명서의 그룹 업데이트를 참조하세요. AWS

예제 2: CloudFormation 스택 기반 리소스 그룹에 대한 쿼리를 업데이트하려면

다음 update-group-query 예제에서는 지정된 AWS CloudFormation 스택 기반 리소스 그룹에 연결된 쿼리를 업데이트합니다.

```
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\"}"}'
```

출력:

```
{
    "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\"}"
    }
}
```

자세한 내용은 리소스 그룹 사용 설명서의 그룹 업데이트를 참조하세요. AWS

- 자세한 API 내용은 명령 참조UpdateGroupQuery의 섹션을 참조하세요. AWS CLI

**update-group**

다음 코드 예시에서는 update-group을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 그룹에 대한 설명을 업데이트하려면

다음 update-group 예제에서는 지정된 리소스 그룹에 대한 설명을 업데이트합니다.

```
aws resource-groups update-group \
    --group-name tbq-WebServer \
    --description "Resource group for all web server resources."
```

출력:

```
{
    "Group": {
        "GroupArn": "arn:aws:resource-groups:us-west-2:123456789012:group/tbq-
WebServer",
        "Name": "tbq-WebServer"
        "Description": "Resource group for all web server resources."
    }
}
```

자세한 내용은 리소스 그룹 사용 설명서의 그룹 업데이트를 참조하세요. AWS

- 자세한 API 내용은 명령 참조UpdateGroup의 섹션을 참조하세요. AWS CLI

## 를 사용하여 리소스 그룹 태그 지정 API 예제 AWS CLI

다음 코드 예제에서는 Resource Groups Tagging과 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다API.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `get-resources`

다음 코드 예시에서는 `get-resources`을 사용하는 방법을 보여 줍니다.

AWS CLI

태그가 지정된 리소스 목록을 가져오려면

다음 `get-resources` 예제에서는 지정된 키 이름 및 값으로 태그가 지정된 계정의 리소스 목록을 표시합니다.

```
aws resourcegroupstaggingapi get-resources \
    --tag-filters Key=Environment,Values=Production \
    --tags-per-page 100
```

출력:

```
{
    "ResourceTagMappingList": [
        {
            "ResourceARN": " arn:aws:inspector:us-west-2:123456789012:target/0-
nvgVhaxX/template/0-7sbz2Kz0",
            "Tags": [
                {
                    "Key": "Environment",
                    "Value": "Production"
                }
            ]
        }
    ]
}
```

자세한 내용은 리소스 그룹 태그 지정 참조GetResources의 섹션을 참조하세요.  API

- 자세한 API 내용은 명령 참조GetResources의 섹션을 참조하세요. AWS CLI

### get-tag-keys

다음 코드 예시에서는 get-tag-keys을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 태그 키 목록을 가져오려면

다음 get-tag-keys 예제에서는 계정의 리소스에서 사용하는 모든 태그 키 이름 목록을 검색합니다.

```
aws resourcegroupstaggingapi get-tag-keys
```

출력:

```
{
    "TagKeys": [
        "Environment",
        "CostCenter",
        "Department"
    ]
}
```

자세한 내용은 리소스 그룹 태그 지정 참조GetTagKeys의 섹션을 참조하세요.  API

- 자세한 API 내용은 명령 참조GetTagKeys의 섹션을 참조하세요. AWS CLI

### get-tag-values

다음 코드 예시에서는 get-tag-values을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 태그 값 목록을 가져오려면

다음 get-tag-values 예제에서는 의 모든 리소스에 대해 지정된 키에 사용된 모든 값을 표시합니다.

```
aws resourcegroupstaggingapi get-tag-values \
    --key=Environment
```

출력:

```
{
    "TagValues": [
        "Alpha",
        "Gamma",
        "Production"
    ]
}
```

자세한 내용은 리소스 그룹 태그 지정 참조GetTagValues의 섹션을 참조하세요. API

- 자세한 API 내용은 명령 참조GetTagValues의 섹션을 참조하세요. AWS CLI

### tag-resources

다음 코드 예시에서는 tag-resources을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 태그를 연결하려면

다음 tag-resources 예제에서는 지정된 리소스에 키 이름과 값을 태그로 지정합니다.

```
aws resourcegroupstaggingapi tag-resources \
    --resource-arn-list arn:aws:s3:::MyProductionBucket \
    --tags Environment=Production,CostCenter=1234
```

출력:

```
{
    "FailedResourcesMap": {}
}
```

자세한 내용은 리소스 그룹 태그 지정 참조TagResources의 섹션을 참조하세요. API

- 자세한 API 내용은 명령 참조TagResources의 섹션을 참조하세요. AWS CLI

### untag-resources

다음 코드 예시에서는 untag-resources을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에서 태그를 제거하려면

다음 `untag-resources` 예제에서는 지정된 리소스에서 지정된 태그 키와 연결된 값을 제거합니다.

```
aws resourcegroupstaggingapi untag-resources \
    --resource-arn-list arn:aws:s3:::awsexamplebucket \
    --tag-keys Environment CostCenter
```

출력:

```
{
    "FailedResourcesMap": {}
}
```

자세한 내용은 리소스 그룹 태그 지정 참조UntagResources의 섹션을 참조하세요.  API

- 자세한 API 내용은 명령 참조UntagResources의 섹션을 참조하세요. AWS CLI

# AWS RoboMaker 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS RoboMaker.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### `batch-describe-simulation-job`

다음 코드 예시에서는 `batch-describe-simulation-job`을 사용하는 방법을 보여 줍니다.

AWS CLI

시뮬레이션 작업을 일괄 설명하는 방법

다음 batch-describe-simulation-job 예제에서는 지정된 세 가지 시뮬레이션 작업에 대한 세부 정보를 검색합니다.

명령:

```
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-
g8h6tglmblgw
```

출력:

```
{
    "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-G0OBWTQ8YBG6",
            "robotApplications": [
                {
```

```
                    "application": "arn:aws:robomaker:us-west-2:111111111111:robot-
application/AWSRoboMakerCloudWatch-1547663411642_NZbpqEJ3T/1547663517377",
                    "applicationVersion": "$LATEST",
                    "launchConfig": {
                        "packageName": "cloudwatch_robot",
                        "launchFile": "await_commands.launch",
                        "environmentVariables": {
                            "LAUNCH_ID": "1548168752173",
                            "ROS_AWS_REGION": "us-west-2"
                        }
                    }
                }
            ],
            "simulationApplications": [
                {
                    "application": "arn:aws:robomaker:us-
west-2:111111111111:simulation-application/
AWSRoboMakerCloudWatch-1547663411642_0LIt6D1h6/1547663521470",
                    "applicationVersion": "$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": []
}
```

- 자세한 API 내용은 명령 참조BatchDescribeSimulationJob의 섹션을 참조하세요. AWS CLI

### cancel-simulation-job

다음 코드 예시에서는 cancel-simulation-job을 사용하는 방법을 보여 줍니다.

AWS CLI

시뮬레이션 작업을 취소하려면

다음 cancel-simulation-job 예제에서는 지정된 시뮬레이션 작업을 취소합니다.

```
aws robomaker cancel-simulation-job \
    --job arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-66bbb3gpxm8x
```

- 자세한 API 내용은 명령 참조CancelSimulationJob의 섹션을 참조하세요. AWS CLI

### create-deployment-job

다음 코드 예시에서는 create-deployment-job을 사용하는 방법을 보여 줍니다.

AWS CLI

배포 작업을 생성하려면

이 예제에서는 플릿 에 대한 배포 작업을 생성합니다 MyFleet. 여기에는 “ENVIRONMENT”이라는 환경 변수가 포함됩니다. 또한 “Region”이라는 태그를 연결합니다.

명령:

```
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
```

출력:

```
{
  "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"
  }
}
```

- 자세한 API 내용은 명령 참조CreateDeploymentJob의 섹션을 참조하세요. AWS CLI

### `create-fleet`

다음 코드 예시에서는 `create-fleet`을 사용하는 방법을 보여 줍니다.

AWS CLI

플릿을 생성하려면

이 예제에서는 플릿을 생성합니다. 리전이라는 태그를 연결합니다.

명령:

```
aws robomaker create-fleet --name MyFleet --tags Region=East
```

출력:

```
{
  "arn": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/
MyOtherFleet/1550771394395",
  "name": "MyFleet",
  "createdAt": 1550771394.0,
  "tags": {
      "Region": "East"
  }
}
```

- 자세한 API 내용은 명령 참조CreateFleet의 섹션을 참조하세요. AWS CLI

### `create-robot-application-version`

다음 코드 예시에서는 `create-robot-application-version`을 사용하는 방법을 보여 줍니다.

AWS CLI

로봇 애플리케이션 버전을 생성하려면

이 예제에서는 로봇 애플리케이션 버전을 생성합니다.

명령:

```
aws robomaker create-robot-application-version --application arn:aws:robomaker:us-
west-2:111111111111:robot-application/MyRobotApplication/1551201873931
```

출력:

```
{
  "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": {}
}
```

- 자세한 API 내용은 명령 참조CreateRobotApplicationVersion의 섹션을 참조하세요. AWS CLI

### create-robot-application

다음 코드 예시에서는 create-robot-application을 사용하는 방법을 보여 줍니다.

AWS CLI

로봇 애플리케이션을 생성하려면

이 예제에서는 로봇 애플리케이션을 생성합니다.

명령:

```
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
```

출력:

```
{
  "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": {}
}
```

- 자세한 API 내용은 명령 참조CreateRobotApplication의 섹션을 참조하세요. AWS CLI

### `create-robot`

다음 코드 예시에서는 `create-robot`을 사용하는 방법을 보여 줍니다.

AWS CLI

로봇을 생성하려면

이 예제에서는 로봇을 생성합니다. ARMHF 아키텍처를 사용합니다. 또한 리전이라는 태그를 연결합니다.

명령:

```
aws robomaker create-robot --name MyRobot --architecture ARMHF --greengrass-group-id 0f728a3c-7dbf-4a3e-976d-d16a8360caba --tags Region=East
```

출력:

```
{
  "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"
  }
}
```

- 자세한 API 내용은 명령 참조CreateRobot의 섹션을 참조하세요. AWS CLI

### create-simulation-application-version

다음 코드 예시에서는 create-simulation-application-version을 사용하는 방법을 보여 줍니다.

AWS CLI

시뮬레이션 애플리케이션 버전을 생성하려면

이 예제에서는 로봇 애플리케이션 버전을 생성합니다.

명령:

```
aws robomaker create-simulation-application-version --
application arn:aws:robomaker:us-west-2:111111111111:robot-application/
MySimulationApplication/1551203427605
```

출력:

```
{
  "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": {}
}
```

- 자세한 API 내용은 명령 참조CreateSimulationApplicationVersion의 섹션을 참조하세요. AWS CLI

### create-simulation-application

다음 코드 예시에서는 create-simulation-application을 사용하는 방법을 보여 줍니다.

AWS CLI

시뮬레이션 애플리케이션을 생성하려면

이 예제에서는 시뮬레이션 애플리케이션을 생성합니다.

명령:

```
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
```

출력:

```
{
  "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": {}
}
```

- 자세한 API 내용은 명령 참조CreateSimulationApplication의 섹션을 참조하세요. AWS CLI

### `create-simulation-job`

다음 코드 예시에서는 `create-simulation-job`을 사용하는 방법을 보여 줍니다.

AWS CLI

시뮬레이션 작업을 생성하려면

이 예제에서는 시뮬레이션 작업을 생성합니다. 로봇 애플리케이션과 시뮬레이션 애플리케이션을 사용합니다.

명령:

```
aws robomaker create-simulation-job --max-job-duration-
in-seconds 3600 --iam-role arn:aws:iam::111111111111:role/
AWSRoboMakerCloudWatch-154766341-SimulationJobRole-G0OBWTQ8YBG6 --robot-
applications application=arn:aws:robomaker:us-west-2:111111111111:robot-application/
MyRobotApplication/1551203485821,launchConfig={packageName=hello_world_robot,launchFile=rota
 --simulation-applications application=arn:aws:robomaker:us-
west-2:111111111111:simulation-application/
MySimulationApplication/1551203427605,launchConfig={packageName=hello_world_simulation,launc
 --tags Region=North
```

출력:

```
{
  "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"
  }
}
```

- 자세한 API 내용은 명령 참조CreateSimulationJob의 섹션을 참조하세요. AWS CLI

### `delete-fleet`

다음 코드 예시에서는 `delete-fleet`을 사용하는 방법을 보여 줍니다.

AWS CLI

플릿을 삭제하려면

이 예제에서는 플릿을 삭제합니다.

명령:

```
aws robomaker delete-fleet --fleet arn:aws:robomaker:us-
west-2:111111111111:deployment-fleet/MyFleet/1550771394395
```

- 자세한 API 내용은 명령 참조DeleteFleet의 섹션을 참조하세요. AWS CLI

### delete-robot-application

다음 코드 예시에서는 delete-robot-application을 사용하는 방법을 보여 줍니다.

AWS CLI

로봇 애플리케이션을 삭제하려면

이 예제에서는 로봇 애플리케이션을 삭제합니다.

명령:

```
aws robomaker delete-robot-application --application arn:aws:robomaker:us-
west-2:111111111111:robot-application/MyRobotApplication/1551203485821
```

- 자세한 API 내용은 명령 참조DeleteRobotApplication의 섹션을 참조하세요. AWS CLI

### delete-robot

다음 코드 예시에서는 delete-robot을 사용하는 방법을 보여 줍니다.

AWS CLI

로봇을 삭제하려면

이 예제에서는 로봇을 삭제합니다.

명령:

```
aws robomaker delete-robot --robot arn:aws:robomaker:us-west-2:111111111111:robot/
MyRobot/1540829698778
```

- 자세한 API 내용은 명령 참조DeleteRobot의 섹션을 참조하세요. AWS CLI

### delete-simulation-application

다음 코드 예시에서는 delete-simulation-application을 사용하는 방법을 보여 줍니다.

AWS CLI

시뮬레이션 애플리케이션을 삭제하려면

이 예제에서는 시뮬레이션 애플리케이션을 삭제합니다.

명령:

```
aws robomaker delete-simulation-application --application arn:aws:robomaker:us-
west-2:111111111111:simulation-application/MySimulationApplication/1551203427605
```

- 자세한 API 내용은 명령 참조DeleteSimulationApplication의 섹션을 참조하세요. AWS CLI

### deregister-robot

다음 코드 예시에서는 deregister-robot을 사용하는 방법을 보여 줍니다.

AWS CLI

플릿에서 로봇 등록을 취소하려면

이 예제는 플릿에서 로봇의 등록을 취소합니다.

명령:

```
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
```

출력:

```
{
  "fleet": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/
MyFleet/1550771358907",
  "robot": "arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1550772324398"
}
```

- 자세한 API 내용은 명령 참조DeregisterRobot의 섹션을 참조하세요. AWS CLI

### describe-deployment-job

다음 코드 예시에서는 describe-deployment-job을 사용하는 방법을 보여 줍니다.

AWS CLI

배포 작업을 설명하려면

다음 describe-deployment-job 예제에서는 지정된 배포 작업에 대한 세부 정보를 검색합니다.

```
aws robomaker describe-deployment-job \
    --job arn:aws:robomaker:us-west-2:111111111111:deployment-job/deployment-xl8qssl6pbcn
```

출력:

```
{
    "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": {}
}
```

- 자세한 API 내용은 명령 참조DescribeDeploymentJob의 섹션을 참조하세요. AWS CLI

### `describe-fleet`

다음 코드 예시에서는 `describe-fleet`을 사용하는 방법을 보여 줍니다.

AWS CLI

플릿을 설명하려면

다음 `describe-fleet` 예제에서는 지정된 플릿에 대한 세부 정보를 검색합니다.

```
aws robomaker describe-fleet \
    --fleet arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/
MyFleet/1550771358907
```

출력:

```
{
    "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": {}
}
```

- 자세한 API 내용은 명령 참조DescribeFleet의 섹션을 참조하세요. AWS CLI

### `describe-robot-application`

다음 코드 예시에서는 `describe-robot-application`을 사용하는 방법을 보여 줍니다.

AWS CLI

로봇 애플리케이션을 설명하려면

이 예제에서는 로봇 애플리케이션에 대해 설명합니다.

명령:

```
aws robomaker describe-robot-application --application arn:aws:robomaker:us-
west-2:111111111111:robot-application/MyRobotApplication/1551203485821
```

출력:

```
{
  "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": {}
}
```

- 자세한 API 내용은 명령 참조DescribeRobotApplication의 섹션을 참조하세요. AWS CLI

### describe-robot

다음 코드 예시에서는 describe-robot을 사용하는 방법을 보여 줍니다.

AWS CLI

로봇을 설명하려면

이 예제에서는 로봇에 대해 설명합니다.

명령:

```
aws robomaker describe-robot --robot arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1550772324398
```

출력:

```
{
  "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"
  }
}
```

- 자세한 API 내용은 명령 참조DescribeRobot의 섹션을 참조하세요. AWS CLI

### describe-simulation-application

다음 코드 예시에서는 describe-simulation-application을 사용하는 방법을 보여 줍니다.

AWS CLI

시뮬레이션 애플리케이션을 설명하려면

이 예제에서는 시뮬레이션 애플리케이션에 대해 설명합니다.

명령:

```
aws robomaker describe-simulation-application --application arn:aws:robomaker:us-west-2:111111111111:simulation-application/MySimulationApplication/1551203427605
```

출력:

```
{
  "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": {}
}
```

- 자세한 API 내용은 명령 참조DescribeSimulationApplication의 섹션을 참조하세요. AWS CLI

### describe-simulation-job

다음 코드 예시에서는 describe-simulation-job을 사용하는 방법을 보여 줍니다.

AWS CLI

시뮬레이션 작업을 설명하려면

이 예제에서는 시뮬레이션 작업에 대해 설명합니다.

명령:

```
aws robomaker describe-simulation-job --job arn:aws:robomaker:us-
west-2:111111111111:simulation-job/sim-pql32v7pfjy6
```

출력:

```
{
  "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": {}
}
```

- 자세한 API 내용은 명령 참조DescribeSimulationJob의 섹션을 참조하세요. AWS CLI

### `list-deployment-jobs`

다음 코드 예시에서는 `list-deployment-jobs`을 사용하는 방법을 보여 줍니다.

AWS CLI

배포 작업을 나열하려면

다음 `list-deployment-jobs` 예제에서는 배포 작업 목록을 검색합니다.

```
aws robomaker list-deployment-jobs
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조[ListDeploymentJobs](#)의 섹션을 참조하세요. AWS CLI

### list-fleets

다음 코드 예시에서는 list-fleets을 사용하는 방법을 보여 줍니다.

AWS CLI

플릿을 나열하려면

이 예제에서는 플릿을 나열합니다. 최대 20대의 플릿이 반환됩니다.

명령:

```
aws robomaker list-fleets --max-items 20
```

출력:

```
{
  "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
      }
  ]
}
```

- 자세한 API 내용은 명령 참조ListFleets의 섹션을 참조하세요. AWS CLI

### list-robot-applications

다음 코드 예시에서는 list-robot-applications을 사용하는 방법을 보여 줍니다.

AWS CLI

로봇 애플리케이션을 나열하려면

이 예제에서는 로봇 애플리케이션을 나열합니다. 결과는 로봇 애플리케이션 20개로 제한됩니다.

명령:

```
aws robomaker list-robot-applications --max-results 20
```

출력:

```
{
  "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
      }
  ]
}
```

- 자세한 API 내용은 명령 참조ListRobotApplications의 섹션을 참조하세요. AWS CLI

### `list-robots`

다음 코드 예시에서는 `list-robots`을 사용하는 방법을 보여 줍니다.

AWS CLI

로봇을 나열하려면

이 예제에서는 로봇을 나열합니다. 최대 20개의 로봇이 반환됩니다.

명령:

```
aws robomaker list-robots --max-results 20
```

출력:

```
{
  "robots": [
      {
          "arn": "arn:aws:robomaker:us-west-2:111111111111:robot/
Robot100/1544035373264",
          "name": "Robot100",
          "status": "Available",
          "createdAt": 1544035373.0,
          "architecture": "X86_64"
      },
      {
          "arn": "arn:aws:robomaker:us-west-2:111111111111:robot/
Robot101/1542146976587",
          "name": "Robot101",
          "status": "Available",
          "createdAt": 1542146976.0,
          "architecture": "X86_64"
      },
      {
          "arn": "arn:aws:robomaker:us-west-2:111111111111:robot/
Robot102/1540834232469",
          "name": "Robot102",
          "fleetArn": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/
Trek/1539894765711",
          "status": "Available",
          "createdAt": 1540834232.0,
          "architecture": "X86_64",
          "lastDeploymentJob": "arn:aws:robomaker:us-west-2:111111111111:deployment-
job/deployment-jb007b75gl5f",
          "lastDeploymentTime": 1550689533.0
      },
      {
          "arn": "arn:aws:robomaker:us-west-2:111111111111:robot/
MyRobot/1540829698778",
          "name": "MyRobot",
```

```
            "status": "Registered",
            "createdAt": 1540829698.0,
            "architecture": "X86_64"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListRobots의 섹션을 참조하세요. AWS CLI

### `list-simulation-applications`

다음 코드 예시에서는 `list-simulation-applications`을 사용하는 방법을 보여 줍니다.

AWS CLI

시뮬레이션 애플리케이션을 나열하려면

이 예제에서는 시뮬레이션 애플리케이션을 나열합니다. 최대 20개의 시뮬레이션 애플리케이션이 반환됩니다.

명령:

```
aws robomaker list-simulation-applications --max-results 20
```

출력:

```
{
  "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_0LIt6D1h6",
            "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-application/
AWSRoboMakerCloudWatch-1547663411642_0LIt6D1h6/1547663521470",
            "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListSimulationApplications의 섹션을 참조하세요. AWS CLI

### `list-simulation-jobs`

다음 코드 예시에서는 `list-simulation-jobs`을 사용하는 방법을 보여 줍니다.

AWS CLI

시뮬레이션 작업을 나열하려면

이 예제에서는 시뮬레이션 작업을 나열합니다.

명령:

```
aws robomaker list-simulation-jobs
```

출력:

```
{
  "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-
b27c4rkrtzcw",
          "lastUpdatedAt": 1543514088.0,
          "status": "Canceled",
          "simulationApplicationNames": [
              "AWSRoboMakerPersonDetection-1543513948280_T8rHW2_lu"
          ],
          "robotApplicationNames": [
              "AWSRoboMakerPersonDetection-1543513948280_EYaMT0mYb"
          ]
      },
      {
          "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/
sim-51vxjbzy4q8t",
          "lastUpdatedAt": 1543508858.0,
          "status": "Canceled",
          "simulationApplicationNames": [
              "AWSRoboMakerCloudWatch-1543504747391_lFF9ZQyx6"
          ],
          "robotApplicationNames": [
              "AWSRoboMakerCloudWatch-1543504747391_axbYa3S3K"
          ]
      },
      {
```

```
            "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-
kgf1fqxflqbx",
            "lastUpdatedAt": 1543504862.0,
            "status": "Completed",
            "simulationApplicationNames": [
                "AWSRoboMakerCloudWatch-1543504747391_lFF9ZQyx6"
            ],
            "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_Jg96l"
            ],
            "robotApplicationNames": [
                "AWSRoboMakerHelloWorld-1543437372341_lNbmKHvs9"
            ]
        },
        {
            "arn": "arn:aws:robomaker:us-west-2:111111111111:simulation-job/sim-
txy5ypxmhz84",
            "lastUpdatedAt": 1543437488.0,
            "status": "Completed",
            "simulationApplicationNames": [
                "AWSRoboMakerHelloWorld-1543437372341__yb_Jg96l"
            ],
            "robotApplicationNames": [
                "AWSRoboMakerHelloWorld-1543437372341_lNbmKHvs9"
            ]
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListSimulationJobs의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스의 태그를 나열하려면

이 예제에서는 AWS RoboMaker 리소스에 대한 태그를 나열합니다.

명령:

```
aws robomaker list-tags-for-resource --resource-arn "arn:aws:robomaker:us-west-2:111111111111:robot/Robby_the_Robot/1544035373264"
```

출력:

```
{
  "tags": {
      "Region": "North",
      "Stage": "Initial"
  }
}
```

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

## register-robot

다음 코드 예시에서는 register-robot을 사용하는 방법을 보여 줍니다.

AWS CLI

로봇을 등록하려면

이 예제에서는 로봇을 플릿에 등록합니다.

명령:

```
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
```

출력:

```
{
```

```
  "fleet": "arn:aws:robomaker:us-west-2:111111111111:deployment-fleet/
MyFleet/1550771358907",
  "robot": "arn:aws:robomaker:us-west-2:111111111111:robot/MyRobot/1550772324398"
}
```

- 자세한 API 내용은 명령 참조RegisterRobot의 섹션을 참조하세요. AWS CLI

### `restart-simulation-job`

다음 코드 예시에서는 `restart-simulation-job`을 사용하는 방법을 보여 줍니다.

AWS CLI

시뮬레이션을 다시 시작하려면

이 예제에서는 시뮬레이션을 다시 시작합니다.

명령:

```
aws robomaker restart-simulation-job --job arn:aws:robomaker:us-
west-2:111111111111:simulation-job/sim-t6rdgt70mftr
```

- 자세한 API 내용은 명령 참조RestartSimulationJob의 섹션을 참조하세요. AWS CLI

### `sync-deployment-job`

다음 코드 예시에서는 `sync-deployment-job`을 사용하는 방법을 보여 줍니다.

AWS CLI

배포 작업을 동기화하려면

이 예제에서는 배포 작업을 동기화합니다.

명령:

```
aws robomaker sync-deployment-job --fleet arn:aws:robomaker:us-
west-2:111111111111:deployment-fleet/Trek/1539894765711
```

출력:

```
{
  "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
}
```

- 자세한 API 내용은 명령 참조SyncDeploymentJob의 섹션을 참조하세요. AWS CLI

### `tag-resource`

다음 코드 예시에서는 `tag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 태그를 지정하려면

이 예제에서는 리소스에 태그를 지정합니다. 리전과 스테이지라는 두 개의 태그를 연결합니다.

명령:

```
aws robomaker tag-resource --resource-arn "arn:aws:robomaker:us-
west-2:111111111111:robot/MyRobot/1544035373264" --tags Region=North,Stage=Initial
```

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### untag-resource

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스의 태그를 해제하려면

이 예제는 리소스에서 태그를 제거합니다. 리전 태그를 제거합니다.

명령:

```
aws robomaker untag-resource --resource-arn "arn:aws:robomaker:us-
west-2:111111111111:robot/MyRobot/1544035373264" --tag-keys Region
```

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-robot-application

다음 코드 예시에서는 update-robot-application을 사용하는 방법을 보여 줍니다.

AWS CLI

로봇 애플리케이션을 업데이트하려면

이 예제는 로봇 애플리케이션을 업데이트합니다.

명령:

```
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
```

출력:

```
{
  "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"
}
```

- 자세한 API 내용은 명령 참조UpdateRobotApplication의 섹션을 참조하세요. AWS CLI

### update-simulation-application

다음 코드 예시에서는 update-simulation-application을 사용하는 방법을 보여 줍니다.

AWS CLI

시뮬레이션 애플리케이션을 업데이트하려면

이 예제에서는 시뮬레이션 애플리케이션을 업데이트합니다.

명령:

```
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
```

출력:

```
{
  "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"
}
```

- 자세한 API 내용은 명령 참조UpdateSimulationApplication의 섹션을 참조하세요. AWS CLI

## 를 사용하여 Route 53 예제 AWS CLI

다음 코드 예제에서는 Route 53과 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### change-resource-record-sets

다음 코드 예시에서는 change-resource-record-sets을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 레코드 세트를 생성, 업데이트 또는 삭제하려면

다음 change-resource-record-sets 명령은 파일 에서 hosted-zone-id Z1R8UBAEXAMPLE 및 JSON형식 구성을 사용하여 리소스 레코드 세트를 생성합니다C:\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
```

자세한 내용은 Amazon Route 53 참조POST ChangeResourceRecordSets 의 섹션을 참조하세요. Amazon Route 53 API

JSON 파일의 구성은 생성하려는 리소스 레코드 세트의 종류에 따라 달라집니다.

BasicWeightedAliasWeighted AliasLatencyLatency AliasFailoverFailover 별칭

기본 구문:

```
{
  "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"
          },
          {...}
        ]
      }
    },
```

```
    {...}
  ]
}
```

가중 구문:

```
{
  "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"
      }
    },
    {...}
  ]
}
```

별칭 구문:

```
{
  "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"
      }
    },
    {...}
  ]
}
```

가중 별칭 구문:

```
{
  "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"
      }
    },
    {...}
  ]
}
```

지연 시간 구문:

```
{
  "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"
      }
    },
    {...}
  ]
}
```

지연 시간 별칭 구문:

```
{
  "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"
```

```
      }
    },
    {...}
  ]
}
```

장애 조치 구문:

```
{
  "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"
      }
    },
    {...}
  ]
}
```

장애 조치 별칭 구문:

```
{
  "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"
      }
    },
    {...}
  ]
}
```

- 자세한 API 내용은 명령 참조ChangeResourceRecordSets의 섹션을 참조하세요. AWS CLI

### change-tags-for-resource

다음 코드 예시에서는 change-tags-for-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 ID로 지정된 healthcheck 리소스owner에 라는 태그를 추가합니다.

```
aws route53 change-tags-for-resource --resource-type healthcheck --resource-
id 6233434j-18c1-34433-ba8e-3443434 --add-tags Key=owner,Value=myboss
```

다음 명령은 ID로 지정된 호스팅 영역 리소스owner에서 라는 태그를 제거합니다.

```
aws route53 change-tags-for-resource --resource-type hostedzone --resource-
id Z1523434445 --remove-tag-keys owner
```

- 자세한 API 내용은 명령 참조ChangeTagsForResource의 섹션을 참조하세요. AWS CLI

### create-health-check

다음 코드 예시에서는 create-health-check을 사용하는 방법을 보여 줍니다.

AWS CLI

상태 확인을 생성하려면

다음 create-health-check 명령은 호출자 참조2014-04-01-18:47와 파일 의 JSON형식 구성을 사용하여 상태 확인을 생성합니다C:\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
```

JSON 구문:

```
{
  "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
}
```

Route 53 리소스 레코드 세트에 상태 확인을 추가하려면 change-resource-record-sets 명령을 사용합니다.

자세한 내용은 Amazon Route 53 개발자 안내서의 Amazon Route 53 상태 확인 및 DNS 장애 조치를 참조하세요.

- 자세한 API 내용은 명령 참조CreateHealthCheck의 섹션을 참조하세요. AWS CLI

### create-hosted-zone

다음 코드 예시에서는 create-hosted-zone을 사용하는 방법을 보여 줍니다.

AWS CLI

호스팅 영역을 생성하려면

다음 create-hosted-zone 명령은 호출자 참조 를 example.com 사용하여 라는 호스팅 영역을 추가합니다2014-04-01-18:47. 선택적 주석에는 공백이 포함되므로 따옴표로 묶어야 합니다.

```
aws route53 create-hosted-zone --name example.com --caller-
reference 2014-04-01-18:47 --hosted-zone-config Comment="command-line version"
```

자세한 내용은 Amazon Route 53 개발자 안내서의 호스팅 영역 작업을 참조하세요.

- 자세한 API 내용은 명령 참조CreateHostedZone의 섹션을 참조하세요. AWS CLI

### delete-health-check

다음 코드 예시에서는 delete-health-check을 사용하는 방법을 보여 줍니다.

AWS CLI

상태 확인을 삭제하려면

다음 delete-health-check 명령은 health-check-id 의 를 사용하여 상태 확인을 삭제합니다e75b48d9-547a-4c3d-88a5-ae4002397608.

```
aws route53 delete-health-check --health-check-id e75b48d9-547a-4c3d-88a5-
ae4002397608
```

- 자세한 API 내용은 명령 참조DeleteHealthCheck의 섹션을 참조하세요. AWS CLI

### delete-hosted-zone

다음 코드 예시에서는 delete-hosted-zone을 사용하는 방법을 보여 줍니다.

AWS CLI

호스팅 영역을 삭제하려면

다음 delete-hosted-zone 명령은 id 의 를 사용하여 호스팅 영역을 삭제합니다Z36KTIQEXAMPLE.

```
aws route53 delete-hosted-zone --id Z36KTIQEXAMPLE
```

- 자세한 API 내용은 명령 참조DeleteHostedZone의 섹션을 참조하세요. AWS CLI

### get-change

다음 코드 예시에서는 get-change을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 레코드 세트에 대한 변경 상태를 가져오려면

다음 get-change 명령은 의 Id 가 있는 change-resource-record-sets 요청에 대한 상태 및 기타 정보를 가져옵니다/change/CWPIK4URU2I5S.

```
aws route53 get-change --id /change/CWPIK4URU2I5S
```

- 자세한 API 내용은 명령 참조GetChange의 섹션을 참조하세요. AWS CLI

### get-health-check

다음 코드 예시에서는 get-health-check을 사용하는 방법을 보여 줍니다.

AWS CLI

상태 확인에 대한 정보를 가져오려면

다음 get-health-check 명령은 health-check-id 의 가 있는 상태 확인에 대한 정보를 가져옵니다02ec8401-9879-4259-91fa-04e66d094674.

```
aws route53 get-health-check --health-check-id 02ec8401-9879-4259-91fa-04e66d094674
```

- 자세한 API 내용은 명령 참조GetHealthCheck의 섹션을 참조하세요. AWS CLI

### get-hosted-zone

다음 코드 예시에서는 get-hosted-zone을 사용하는 방법을 보여 줍니다.

AWS CLI

호스팅 영역에 대한 정보를 가져오려면

다음 get-hosted-zone 명령은 id 의 를 사용하여 호스팅 영역에 대한 정보를 가져옵니다Z1R8UBAEXAMPLE.

```
aws route53 get-hosted-zone --id Z1R8UBAEXAMPLE
```

- 자세한 API 내용은 명령 참조GetHostedZone의 섹션을 참조하세요. AWS CLI

### `list-health-checks`

다음 코드 예시에서는 `list-health-checks`을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 AWS 계정과 연결된 상태 확인을 나열하려면

다음 `list-health-checks` 명령은 현재 AWS 계정과 연결된 처음 100개의 상태 검사에 대한 자세한 정보를 나열합니다.

```
aws route53 list-health-checks
```

상태 확인이 100개 이상 있거나 100개 미만의 그룹으로 나열하려면 `--maxitems` 파라미터를 포함하세요. 예를 들어 상태 확인을 한 번에 하나씩 나열하려면 다음 명령을 사용합니다.

```
aws route53 list-health-checks --max-items 1
```

다음 상태 확인을 보려면 이전 명령에 대한 응답`NextToken`에서 값을 가져와`--starting-token`서 파라미터에 포함시킵니다. 예를 들어:

```
aws route53 list-health-checks --max-items 1 --starting-token Z3M3LMPEXAMPLE
```

- 자세한 API 내용은 명령 참조ListHealthChecks의 섹션을 참조하세요. AWS CLI

### `list-hosted-zones-by-name`

다음 코드 예시에서는 `list-hosted-zones-by-name`을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 도메인 이름으로 정렬된 최대 100개의 호스팅 영역을 나열합니다.

```
aws route53 list-hosted-zones-by-name
```

출력:

```
{
  "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"
}
```

다음 명령은 로 시작하는 이름으로 정렬된 호스팅 영역을 나열합니다www.example.com.

```
aws route53 list-hosted-zones-by-name --dns-name www.example.com
```

출력:

```
{
  "HostedZones": [
      {
          "ResourceRecordSetCount": 2,
          "CallerReference": "mwunderl20150527-1",
          "Config": {
              "Comment": "test",
              "PrivateZone": false
```

```
            },
            "Id": "/hostedzone/Z3P5QSUBK4POTI",
            "Name": "www.example.com."
        }
    ],
    "DNSName": "www.example.com",
    "IsTruncated": false,
    "MaxItems": "100"
}
```

- 자세한 API 내용은 명령 참조ListHostedZonesByName의 섹션을 참조하세요. AWS CLI

### list-hosted-zones

다음 코드 예시에서는 list-hosted-zones을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 AWS 계정과 연결된 호스팅 영역을 나열하려면

다음 list-hosted-zones 명령은 현재 AWS 계정과 연결된 처음 100개의 호스팅 영역에 대한 요약 정보를 나열합니다.

```
aws route53 list-hosted-zones
```

100개 이상의 호스팅 영역이 있거나 100개 미만의 그룹으로 나열하려면 --max-items 파라미터를 포함합니다. 예를 들어, 호스팅 영역을 한 번에 하나씩 나열하려면 다음 명령을 사용합니다.

```
aws route53 list-hosted-zones --max-items 1
```

다음 호스팅 영역에 대한 정보를 보려면 이전 명령에 대한 응답에서 NextToken의 값을 가져와 --starting-token 파라미터에 포함합니다. 예를 들면 다음과 같습니다.

```
aws route53 list-hosted-zones --max-items 1 --starting-token Z3M3LMPEXAMPLE
```

- 자세한 API 내용은 명령 참조ListHostedZones의 섹션을 참조하세요. AWS CLI

### list-query-logging-configs

다음 코드 예시에서는 list-query-logging-configs을 사용하는 방법을 보여 줍니다.

AWS CLI

쿼리 로깅 구성을 나열하려면

다음 `list-query-logging-configs` 예제에서는 호스팅 영역 에 대한 AWS 계정의 처음 100개 쿼리 로깅 구성에 대한 정보를 나열합니다Z1OX3WQEXAMPLE.

```
aws route53 list-query-logging-configs \
    --hosted-zone-id Z1OX3WQEXAMPLE
```

출력:

```
{
    "QueryLoggingConfigs": [
        {
            "Id": "964ff34e-ae03-4f06-80a2-9683cexample",
            "HostedZoneId": "Z1OX3WQEXAMPLE",
            "CloudWatchLogsLogGroupArn": "arn:aws:logs:us-east-1:111122223333:log-
group:/aws/route53/example.com:*"
        }
    ]
}
```

자세한 내용은 Amazon Route 53 개발자 안내서의 DNS 쿼리 로깅을 참조하세요.

- 자세한 API 내용은 명령 참조ListQueryLoggingConfigs의 섹션을 참조하세요. AWS CLI

## `list-resource-record-sets`

다음 코드 예시에서는 `list-resource-record-sets`을 사용하는 방법을 보여 줍니다.

AWS CLI

호스팅 영역의 리소스 레코드 세트를 나열하려면

다음 `list-resource-record-sets` 명령은 지정된 호스팅 영역의 처음 100개 리소스 레코드 세트에 대한 요약 정보를 나열합니다.

```
aws route53 list-resource-record-sets --hosted-zone-id Z2LD58HEXAMPLE
```

호스팅 영역에 100개 이상의 리소스 레코드 세트가 포함되어 있거나 100개 미만의 그룹으로 나열하려는 경우 `--maxitems` 파라미터를 포함합니다. 예를 들어 리소스 레코드 세트를 한 번에 하나씩 나열하려면 다음 명령을 사용합니다.

```
aws route53 list-resource-record-sets --hosted-zone-id Z2LD58HEXAMPLE --max-items 1
```

호스팅 영역의 다음 리소스 레코드 세트에 대한 정보를 보려면 이전 명령에 대한 응답`NextToken`에서 의 값을 가져와`--starting-token`서 파라미터에 포함시킵니다. 예:

```
aws route53 list-resource-record-sets --hosted-zone-id Z2LD58HEXAMPLE --max-items 1
 --starting-token Z3M3LMPEXAMPLE
```

특정 이름의 모든 리소스 레코드 세트를 보려면 `--query` 파라미터를 사용하여 필터링합니다. 예:

```
aws route53 list-resource-record-sets --hosted-zone-id Z2LD58HEXAMPLE --
query "ResourceRecordSets[?Name == 'example.domain.']"
```

- 자세한 API 내용은 명령 참조ListResourceRecordSets의 섹션을 참조하세요. AWS CLI

## 를 사용하여 Route 53 도메인 등록 예제 AWS CLI

다음 코드 예제에서는 Route 53 도메인 등록과 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `check-domain-availability`

다음 코드 예시에서는 `check-domain-availability`을 사용하는 방법을 보여 줍니다.

AWS CLI

Route 53에 도메인 이름을 등록할 수 있는지 확인하려면

다음 `check-domain-availability` 명령은 Route 53을 사용하여 도메인 이름을 등록할 수 있는지 여부에 `example.com` 대한 정보를 반환합니다.

이 명령은 `us-east-1` 리전에서만 실행됩니다. 기본 리전이 로 설정된 경우 `region` 파라미터를 생략할 `us-east-1`수 있습니다.

```
aws route53domains check-domain-availability \
    --region us-east-1 \
    --domain-name example.com
```

출력:

```
{
    "Availability": "UNAVAILABLE"
}
```

Route 53은 `.com` 및 와 같은 많은 수의 최상위 도메인(TLDs)을 지원하지만 `.jp`사용 가능한 를 모두 지원하지는 않습니다TLDs. 도메인의 가용성을 확인하고 Route 53이 를 지원하지 않는 경우 는 다음 메시지를 TLD `check-domain-availability` 반환합니다.

```
An error occurred (UnsupportedTLD) when calling the CheckDomainAvailability
 operation: <top-level domain> tld is not supported.
```

Route 53에 도메인을 등록할 때 사용할 수 TLDs 있는 의 목록은 Amazon Route 53 개발자 안내서의 Amazon Route 53에 등록할 수 있는 도메인을 참조하세요. Amazon Route 53 Amazon Route 53에 도메인 등록에 대한 자세한 내용은 Amazon Route 53 개발자 안내서의 새 도메인 등록을 참조하세요.

- 자세한 API 내용은 명령 참조CheckDomainAvailability의 섹션을 참조하세요. AWS CLI

### `check-domain-transferability`

다음 코드 예시에서는 `check-domain-transferability`을 사용하는 방법을 보여 줍니다.

AWS CLI

도메인을 Route 53으로 전송할 수 있는지 확인하려면

다음 check-domain-transferability 명령은 도메인 이름을 Route 53으로 전송할 수 있는지 여부에 대한 정보를 반환example.com합니다.

이 명령은 us-east-1 리전에서만 실행됩니다. 기본 리전이 로 설정된 경우 region 파라미터를 생략할 us-east-1수 있습니다.

```
aws route53domains check-domain-transferability \
    --region us-east-1 \
    --domain-name example.com
```

출력:

```
{
    "Transferability": {
        "Transferable": "UNTRANSFERABLE"
    }
}
```

자세한 내용은 Amazon Route 53 개발자 안내서의 Amazon Route 53에 도메인 등록 전송을 참조하세요. Amazon Route 53

- 자세한 API 내용은 명령 참조CheckDomainTransferability의 섹션을 참조하세요. AWS CLI

### delete-tags-for-domain

다음 코드 예시에서는 delete-tags-for-domain을 사용하는 방법을 보여 줍니다.

AWS CLI

도메인의 태그를 삭제하려면

다음 delete-tags-for-domain 명령은 지정된 도메인에서 세 개의 태그를 삭제합니다. 태그 값이 아닌 태그 키만 지정합니다.

이 명령은 us-east-1 리전에서만 실행됩니다. 기본 리전이 로 설정된 경우 region 파라미터를 생략할 us-east-1수 있습니다.

```
aws route53domains delete-tags-for-domain \
    --region us-east-1 \
    --domain-name example.com \
    --tags-to-delete accounting-key hr-key engineering-key
```

이 명령은 출력을 생성하지 않습니다.

태그가 삭제되었는지 확인하기 위해 list-tags-for-domain 를 실행할 수 있습니다. 자세한 내용은 Amazon Route 53 개발자 안내서의 Amazon Route 53 리소스 태그 지정을 참조하세요. Amazon Route 53

- 자세한 API 내용은 명령 참조DeleteTagsForDomain의 섹션을 참조하세요. AWS CLI

### `disable-domain-auto-renew`

다음 코드 예시에서는 `disable-domain-auto-renew`을 사용하는 방법을 보여 줍니다.

AWS CLI

도메인의 자동 갱신을 비활성화하려면

다음 `disable-domain-auto-renew` 명령은 도메인에 대한 등록이 만료되기 `example.com` 전에 도메인을 자동으로 갱신하지 않도록 Route 53을 구성합니다.

이 명령은 `us-east-1` 리전에서만 실행됩니다. 기본 리전이 로 설정된 경우 `region` 파라미터를 생략할 `us-east-1`수 있습니다.

```
aws route53domains disable-domain-auto-renew \
    --region us-east-1 \
    --domain-name example.com
```

이 명령은 출력을 생성하지 않습니다.

설정이 변경되었는지 확인하기 위해 get-domain-detail 를 실행할 수 있습니다. 자동 갱신이 비활성화된 경우 의 값은 `AutoRenew`입니다`False`. 자동 갱신에 대한 자세한 내용은 Amazon Route 53 개발자 안내서의 도메인 등록 갱신 <https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/domain-renew.html을 참조하세요.

- 자세한 API 내용은 명령 참조DisableDomainAutoRenew의 섹션을 참조하세요. AWS CLI

### `disable-domain-transfer-lock`

다음 코드 예시에서는 `disable-domain-transfer-lock`을 사용하는 방법을 보여 줍니다.

AWS CLI

도메인에서 전송 잠금을 비활성화하려면

다음 disable-domain-transfer-lock 명령은 도메인의 전송 잠금을 제거하여 도메인을 다른 등록 기관으로 전송할 수 example.com 있습니다. 이 명령은 clientTransferProhibited 상태를 변경합니다.

이 명령은 us-east-1 리전에서만 실행됩니다. 기본 리전이 로 설정된 경우 region 파라미터를 생략할 us-east-1수 있습니다.

```
aws route53domains disable-domain-transfer-lock \
    --region us-east-1 \
    --domain-name example.com
```

출력:

```
{
    "OperationId": "3f28e0ac-126a-4113-9048-cc930example"
}
```

전송 잠금이 변경되었는지 확인하기 위해 get-domain-detail 를 실행할 수 있습니다. 전송 잠금이 비활성화되면 의 값에 가 포함되지 StatusList 않습니다clientTransferProhibited.

전송 프로세스에 대한 자세한 내용은 Amazon Route 53 개발자 안내서의 Amazon Route 53에서 다른 레지스트리로 도메인 전송을 참조하세요. Amazon Route 53

- 자세한 API 내용은 명령 참조DisableDomainTransferLock의 섹션을 참조하세요. AWS CLI

### enable-domain-auto-renew

다음 코드 예시에서는 enable-domain-auto-renew을 사용하는 방법을 보여 줍니다.

AWS CLI

도메인의 자동 갱신을 활성화하려면

다음 enable-domain-auto-renew 명령은 도메인에 대한 등록이 만료되기 example.com 전에 도메인을 자동으로 갱신하도록 Route 53을 구성합니다.

이 명령은 us-east-1 리전에서만 실행됩니다. 기본 리전이 로 설정된 경우 region 파라미터를 생략할 us-east-1수 있습니다.

```
aws route53domains enable-domain-auto-renew \
    --region us-east-1 \
```

```
    --domain-name example.com
```

이 명령은 출력을 생성하지 않습니다. 설정이 변경되었는지 확인하려면 get-domain-detail 를 실행할 수 있습니다. 자동 갱신이 활성화된 경우 의 값은 AutoRenew입니다True.

자동 갱신에 대한 자세한 내용은 Amazon Route 53 개발자 안내서의 도메인 등록 갱신 <https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/domain-renew.html을 참조하세요.

- 자세한 API 내용은 명령 참조EnableDomainAutoRenew의 섹션을 참조하세요. AWS CLI

### enable-domain-transfer-lock

다음 코드 예시에서는 enable-domain-transfer-lock을 사용하는 방법을 보여 줍니다.

AWS CLI

도메인에서 전송 잠금을 활성화하려면

다음 enable-domain-transfer-lock 명령은 지정된 도메인을 잠가 다른 등록 기관으로 전송할 수 없도록 합니다. 이 명령은 clientTransferProhibited 상태를 변경합니다.

이 명령은 us-east-1 리전에서만 실행됩니다. 기본 리전이 로 설정된 경우 region 파라미터를 생략할 us-east-1수 있습니다.

```
aws route53domains enable-domain-transfer-lock \
    --region us-east-1 \
    --domain-name example.com
```

출력:

```
{
    "OperationId": "3f28e0ac-126a-4113-9048-cc930example"
}
```

전송 잠금이 변경되었는지 확인하려면 get-domain-detail 를 실행할 수 있습니다. 전송 잠금이 활성화되면 의 값에 가 StatusList 포함됩니다clientTransferProhibited.

전송 프로세스에 대한 자세한 내용은 Amazon Route 53 개발자 안내서의 Amazon Route 53에서 다른 레지스트리로 도메인 전송을 참조하세요. Amazon Route 53

- 자세한 API 내용은 명령 참조EnableDomainTransferLock의 섹션을 참조하세요. AWS CLI

### get-contact-reachability-status

다음 코드 예시에서는 get-contact-reachability-status을 사용하는 방법을 보여 줍니다.

AWS CLI

등록자 연락 담당자가 확인 이메일에 응답했는지 확인하려면

다음 get-contact-reachability-status 명령은 지정된 도메인의 등록자 연락 담당자가 확인 이메일에 응답했는지 여부에 대한 정보를 반환합니다.

이 명령은 us-east-1 리전에서만 실행됩니다. 기본 리전이 로 설정된 경우 region 파라미터를 생략할 us-east-1수 있습니다.

```
aws route53domains get-contact-reachability-status \
    --region us-east-1 \
    --domain-name example.com
```

출력:

```
{
    "domainName": "example.com",
    "status": "DONE"
}
```

자세한 내용은 Amazon Route 53 개발자 안내서의 승인 및 확인 이메일 재전송을 참조하세요.

- 자세한 API 내용은 명령 참조GetContactReachabilityStatus의 섹션을 참조하세요. AWS CLI

### get-domain-detail

다음 코드 예시에서는 get-domain-detail을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 도메인에 대한 자세한 정보를 가져오려면

다음 get-domain-detail 명령은 지정된 도메인에 대한 자세한 정보를 표시합니다.

이 명령은 us-east-1 리전에서만 실행됩니다. 기본 리전이 로 설정된 경우 region 파라미터를 생략할 us-east-1수 있습니다.

```
aws route53domains get-domain-detail \
    --region us-east-1 \
    --domain-name example.com
```

출력:

```
{
    "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"
    ]
}
```

- 자세한 API 내용은 명령 참조GetDomainDetail의 섹션을 참조하세요. AWS CLI

### get-domain-suggestions

다음 코드 예시에서는 get-domain-suggestions을 사용하는 방법을 보여 줍니다.

AWS CLI

제안된 도메인 이름 목록을 가져오려면

다음 get-domain-suggestions 명령은 도메인 이름 에 따라 제안된 도메인 이름 목록을 표시합니다example.com. 응답에는 사용 가능한 도메인 이름만 포함됩니다. 이 명령은 us-east-1 리전에서만 실행됩니다. 기본 리전이 로 설정된 경우 region 파라미터를 생략할 us-east-1수 있습니다.

```
aws route53domains get-domain-suggestions \
    --region us-east-1 \
    --domain-name example.com \
    --suggestion-count 10 \
    --only-available
```

출력:

```
{
    "SuggestionsList": [
        {
            "DomainName": "egzaampal.com",
            "Availability": "AVAILABLE"
        },
        {
            "DomainName": "examplelaw.com",
            "Availability": "AVAILABLE"
        },
        {
            "DomainName": "examplehouse.net",
            "Availability": "AVAILABLE"
        },
        {
            "DomainName": "homeexample.net",
            "Availability": "AVAILABLE"
        },
        {
            "DomainName": "examplelist.com",
            "Availability": "AVAILABLE"
```

```
        },
        {
            "DomainName": "examplenews.net",
            "Availability": "AVAILABLE"
        },
        {
            "DomainName": "officeexample.com",
            "Availability": "AVAILABLE"
        },
        {
            "DomainName": "exampleworld.com",
            "Availability": "AVAILABLE"
        },
        {
            "DomainName": "exampleart.com",
            "Availability": "AVAILABLE"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetDomainSuggestions의 섹션을 참조하세요. AWS CLI

### get-operation-detail

다음 코드 예시에서는 get-operation-detail을 사용하는 방법을 보여 줍니다.

AWS CLI

작업의 현재 상태를 가져오려면

일부 도메인 등록 작업은 비동기적으로 작동하고 응답을 완료하기 전에 반환합니다. 이러한 작업은 현재 상태를 가져오는 데 사용할 수 있는 작업 ID를 반환합니다. 다음 get-operation-detail 명령은 지정된 작업의 상태를 반환합니다.

이 명령은 us-east-1 리전에서만 실행됩니다. 기본 리전이 로 설정된 경우 region 파라미터를 생략할 us-east-1수 있습니다.

```
aws route53domains get-operation-detail \
    --region us-east-1 \
    --operation-id edbd8d63-7fe7-4343-9bc5-54033example
```

출력:

```
{
    "OperationId": "edbd8d63-7fe7-4343-9bc5-54033example",
    "Status": "SUCCESSFUL",
    "DomainName": "example.com",
    "Type": "DOMAIN_LOCK",
    "SubmittedDate": 1573749367.864
}
```

- 자세한 API 내용은 명령 참조GetOperationDetail의 섹션을 참조하세요. AWS CLI

### `list-domains`

다음 코드 예시에서는 `list-domains`을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 AWS 계정에 등록된 도메인을 나열하려면

다음 `list-domains` 명령은 현재 AWS 계정에 등록된 도메인에 대한 요약 정보를 나열합니다.

이 명령은 `us-east-1` 리전에서만 실행됩니다. 기본 리전이 로 설정된 경우 `region` 파라미터를 생략할 `us-east-1`수 있습니다.

```
aws route53domains list-domains
    --region us-east-1
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListDomains의 섹션을 참조하세요. AWS CLI

### `list-operations`

다음 코드 예시에서는 `list-operations`을 사용하는 방법을 보여 줍니다.

AWS CLI

작업 ID를 반환하는 작업의 상태를 나열하려면

일부 도메인 등록 작업은 비동기적으로 실행되고 응답을 완료하기 전에 반환합니다. 이러한 작업은 현재 상태를 가져오는 데 사용할 수 있는 작업 ID를 반환합니다. 다음 `list-operations` 명령은 현재 도메인 등록 작업에 대한 상태를 포함한 요약 정보를 나열합니다.

이 명령은 `us-east-1` 리전에서만 실행됩니다. 기본 리전이 로 설정된 경우 `region` 파라미터를 생략할 `us-east-1`수 있습니다.

```
aws route53domains list-operations
    --region us-east-1
```

출력:

```
{
    "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
        }
    ]
}
```

출력에는 작업 ID를 반환하고 현재 AWS 계정을 사용하여 등록한 적이 있는 모든 도메인에서 수행한 모든 작업이 포함됩니다. 지정된 날짜 이후에 제출한 작업만 가져오려면 `submitted-since` 파라미터를 포함하고 날짜를 Unix 형식과 Coordinated Universal Time()으로 지정할 수 있습니다 UTC. 다음 명령은 2020년 1UTC월 1일 오전 12시 이후에 제출된 모든 작업의 상태를 가져옵니다.

```
aws route53domains list-operations \
    --submitted-since 1577836800
```

- 자세한 API 내용은 명령 참조ListOperations의 섹션을 참조하세요. AWS CLI

### `list-tags-for-domain`

다음 코드 예시에서는 `list-tags-for-domain`을 사용하는 방법을 보여 줍니다.

AWS CLI

도메인의 태그를 나열하려면

다음 `list-tags-for-domain` 명령은 현재 지정된 도메인과 연결된 태그를 나열합니다.

```
aws route53domains list-tags-for-domain --domain-name example.com
```

이 명령은 `us-east-1` 리전에서만 실행됩니다. 기본 리전이 로 설정된 경우 `region` 파라미터를 생략할 `us-east-1`수 있습니다.

```
aws route53domains list-tags-for-domain \
    --region us-east-1 \
    --domain-name example.com
```

출력:

```
{
    "TagList": [
        {
            "Key": "key1",
            "Value": "value1"
        },
        {
            "Key": "key2",
            "Value": "value2"
        }
    ]
}
```

자세한 내용은 Amazon Route 53 개발자 안내서의 Amazon Route 53 리소스 태그 지정을 참조하세요. Amazon Route 53

- 자세한 API 내용은 명령 참조ListTagsForDomain의 섹션을 참조하세요. AWS CLI

### register-domain

다음 코드 예시에서는 register-domain을 사용하는 방법을 보여 줍니다.

AWS CLI

도메인을 등록하려면

다음 register-domain 명령은 도메인을 등록하고 JSON형식이 지정된 파일에서 모든 파라미터 값을 검색합니다.

이 명령은 us-east-1 리전에서만 실행됩니다. 기본 리전이 로 설정된 경우 region 파라미터를 생략할 us-east-1수 있습니다.

```
aws route53domains register-domain \
    --region us-east-1 \
    --cli-input-json file://register-domain.json
```

register-domain.json의 콘텐츠:

```
{
    "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
}
```

출력:

```
{
    "OperationId": "b114c44a-9330-47d1-a6e8-a0b11example"
}
```

작업이 성공했는지 확인하기 위해 를 실행할 수 있습니다`get-operation-detail`. 자세한 내용은 get-operation-detail 섹션을 참조하세요.

자세한 내용은 Amazon Route 53 개발자 안내서의 새 도메인 등록을 참조하세요.

에 값이 필요한 최상위 도메인(TLDs)`ExtraParams`과 유효한 값에 대한 자세한 내용은 Amazon Route 53 참조ExtraParam의 섹션을 참조하세요. Amazon Route 53 API

- 자세한 API 내용은 명령 참조RegisterDomain의 섹션을 참조하세요. AWS CLI

### `renew-domain`

다음 코드 예시에서는 `renew-domain`을 사용하는 방법을 보여 줍니다.

AWS CLI

도메인을 갱신하려면

다음 `renew-domain` 명령은 지정된 도메인을 5년 동안 갱신합니다. 의 값을 가져오려면 `get-domain-detail` 명령을 `current-expiry-year`사용하고 Unix 형식`ExpirationDate`에서 값을 변환합니다.

이 명령은 `us-east-1` 리전에서만 실행됩니다. 기본 리전이 로 설정된 경우 `region` 파라미터를 생략할 `us-east-1`수 있습니다.

```
aws route53domains renew-domain \
    --region us-east-1 \
    --domain-name example.com \
    --duration-in-years 5 \
```

```
    --current-expiry-year 2020
```

출력:

```
{
    "OperationId": "3f28e0ac-126a-4113-9048-cc930example"
}
```

작업이 성공했는지 확인하기 위해 를 실행할 수 있습니다get-operation-detail. 자세한 내용은 get-operation-detail를 참조하세요.

.com 또는 .org와 같은 각 최상위 도메인(TLD)의 레지스트리는 도메인을 갱신할 수 있는 최대 연도를 제어합니다. 도메인의 최대 갱신 기간을 얻으려면 Amazon Route 53 개발자 안내서의 Amazon Route 53에 등록할 수 있는 도메인TLD에 대한 '등록 및 갱신 기간' 섹션을 참조하세요. Amazon Route 53

자세한 내용은 Amazon Route 53 개발자 안내서의 도메인 등록 갱신을 참조하세요.

- 자세한 API 내용은 명령 참조RenewDomain의 섹션을 참조하세요. AWS CLI

### resend-contact-reachability-email

다음 코드 예시에서는 resend-contact-reachability-email을 사용하는 방법을 보여 줍니다.

AWS CLI

등록자 연락의 현재 이메일 주소로 확인 이메일을 재전송하려면

다음 resend-contact-reachability-email 명령은 example.com 도메인의 등록자 연락처에 대한 현재 이메일 주소로 확인 이메일을 재전송합니다.

이 명령은 us-east-1 리전에서만 실행됩니다. 기본 리전이 로 설정된 경우 region 파라미터를 생략할 us-east-1수 있습니다.

```
aws route53domains resend-contact-reachability-email \
    --region us-east-1 \
    --domain-name example.com
```

출력:

```
{
```

```
    "domainName": "example.com",
    "emailAddress": "moliveira@example.com",
    "isAlreadyVerified": true
}
```

이 예제에서와 `true`같이 값이 `isAlreadyVerified`인 경우 등록자 연락처는 지정된 이메일 주소에 연결할 수 있음을 이미 확인했습니다.

자세한 내용은 Amazon Route 53 개발자 안내서의 권한 부여 및 확인 이메일 재전송을 참조하세요.

- 자세한 API 내용은 명령 참조ResendContactReachabilityEmail의 섹션을 참조하세요. AWS CLI

### `retrieve-domain-auth-code`

다음 코드 예시에서는 `retrieve-domain-auth-code`을 사용하는 방법을 보여 줍니다.

AWS CLI

도메인을 다른 등록 기관으로 전송할 수 있도록 도메인에 대한 권한 부여 코드를 가져오려면

다음 `retrieve-domain-auth-code` 명령은 example.com 도메인의 현재 권한 부여 코드를 가져옵니다. 도메인을 해당 등록 기관으로 전송하려는 경우 이 값을 다른 도메인 등록 기관으로 지정합니다.

이 명령은 `us-east-1` 리전에서만 실행됩니다. 기본 리전이 로 설정된 경우 `region` 파라미터를 생략할 `us-east-1`수 있습니다.

```
aws route53domains retrieve-domain-auth-code \
    --region us-east-1 \
    --domain-name example.com
```

출력:

```
{
    "AuthCode": ")o!v3dJeXampLe"
}
```

자세한 내용은 Amazon Route 53 개발자 안내서의 Amazon Route 53에서 다른 레지스트리로 도메인 전송을 참조하세요. Amazon Route 53

- 자세한 API 내용은 명령 참조RetrieveDomainAuthCode의 섹션을 참조하세요. AWS CLI

**transfer-domain**

다음 코드 예시에서는 transfer-domain을 사용하는 방법을 보여 줍니다.

AWS CLI

도메인을 Amazon Route 53으로 전송하려면

다음 transfer-domain 명령은 JSON형식이 지정된 파일 에서 제공하는 파라미터와 함께 도메인을 Route 53으로 전송합니다C:\temp\transfer-domain.json.

이 명령은 us-east-1 리전에서만 실행됩니다. 기본 리전이 로 설정된 경우 region 파라미터를 생략할 us-east-1수 있습니다.

```
aws route53domains transfer-domain \
    --region us-east-1 \
    --cli-input-json file://C:\temp\transfer-domain.json
```

transfer-domain.json의 콘텐츠:

```
{
    "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
}
```

출력:

```
{
```

```
    "OperationId": "b114c44a-9330-47d1-a6e8-a0b11example"
}
```

작업이 성공했는지 확인하기 위해 를 실행할 수 있습니다get-operation-detail. 자세한 내용은 get-operation-detail 섹션을 참조하세요.

자세한 내용은 Amazon Route 53 개발자 안내서의 Amazon Route 53에 도메인 등록 전송을 참조하세요. Amazon Route 53

- 자세한 API 내용은 명령 참조TransferDomain의 섹션을 참조하세요. AWS CLI

### update-domain-contact-privacy

다음 코드 예시에서는 update-domain-contact-privacy을 사용하는 방법을 보여 줍니다.

AWS CLI

도메인의 연락처에 대한 개인 정보 보호 설정을 업데이트하려면

다음 update-domain-contact-privacy 명령은 example.com 도메인의 관리 연락처에 대한 프라이버시 보호를 끕니다. 이 명령은 us-east-1 리전에서만 실행됩니다.

기본 리전이 로 설정된 경우 region 파라미터를 생략할 us-east-1수 있습니다.

```
aws route53domains update-domain-contact-privacy \
    --region us-east-1 \
    --domain-name example.com \
    --no-admin-privacy
```

출력:

```
{
    "OperationId": "b3a219e9-d801-4244-b533-b7256example"
}
```

작업이 성공했는지 확인하기 위해 를 실행할 수 있습니다get-operation-detail. 자세한 내용은 get-operation-detail 섹션을 참조하세요.

자세한 내용은 Amazon Route 53 개발자 안내서의 도메인에 대한 연락처 정보에 대한 프라이버시 보호 활성화 또는 비활성화를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateDomainContactPrivacy의 섹션을 참조하세요. AWS CLI

### update-domain-contact

다음 코드 예시에서는 update-domain-contact을 사용하는 방법을 보여 줍니다.

AWS CLI

도메인의 연락처 정보를 업데이트하려면

다음 update-domain-contact 명령은 도메인의 연락처 정보를 업데이트하여 JSON형식이 지정된 파일 에서 파라미터를 가져옵니다C:\temp\update-domain-contact.json.

이 명령은 us-east-1 리전에서만 실행됩니다. 기본 리전이 로 설정된 경우 region 파라미터를 생략할 us-east-1수 있습니다.

```
aws route53domains update-domain-contact \
    --region us-east-1 \
    --cli-input-json file://C:\temp\update-domain-contact.json
```

update-domain-contact.json의 콘텐츠:

```
{
    "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"
    }
}
```

출력:

```
{
    "OperationId": "b3a219e9-d801-4244-b533-b7256example"
}
```

작업이 성공했는지 확인하기 위해 get-domain-detail 를 실행할 수 있습니다. 자세한 내용은 Amazon Route 53 개발자 안내서의 도메인에 대한 연락처 정보 업데이트를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateDomainContact의 섹션을 참조하세요. AWS CLI

### update-domain-nameservers

다음 코드 예시에서는 update-domain-nameservers을 사용하는 방법을 보여 줍니다.

AWS CLI

도메인의 이름 서버를 업데이트하려면

다음 update-domain-nameservers 명령은 도메인의 이름 서버를 업데이트합니다.

이 명령은 us-east-1 리전에서만 실행됩니다. 기본 리전이 로 설정된 경우 region 파라미터를 생략할 us-east-1수 있습니다.

```
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=n
```

출력:

```
{
    "OperationId": "f1691ec4-0e7a-489e-82e0-b19d3example"
}
```

작업이 성공했는지 확인하기 위해 get-domain-detail 를 실행할 수 있습니다.

자세한 내용은 Amazon Route 53 개발자 안내서의 도메인에 대한 이름 서버 및 Glue 레코드 추가 또는 변경을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateDomainNameservers의 섹션을 참조하세요. AWS CLI

### update-tags-for-domain

다음 코드 예시에서는 update-tags-for-domain을 사용하는 방법을 보여 줍니다.

AWS CLI

도메인에 대한 태그를 추가하거나 업데이트하려면

다음 update-tags-for-domain 명령은 두 개의 키와 example.com 도메인에 해당하는 값을 추가하거나 업데이트합니다. 키 값을 업데이트하려면 키와 새 값을 포함하면 됩니다. 한 번에 하나의 도메인에서만 태그를 추가하거나 업데이트할 수 있습니다.

이 명령은 us-east-1 리전에서만 실행됩니다. 기본 리전이 로 설정된 경우 region 파라미터를 생략할 us-east-1수 있습니다.

```
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"
```

이 명령은 출력을 생성하지 않습니다. 태그가 추가 또는 업데이트되었는지 확인하려면 list-tags-for-domain 를 실행할 수 있습니다.

자세한 내용은 Amazon Route 53 개발자 안내서의 Amazon Route 53 리소스 태그 지정을 참조하세요. Amazon Route 53

- 자세한 API 내용은 명령 참조UpdateTagsForDomain의 섹션을 참조하세요. AWS CLI

### `view-billing`

다음 코드 예시에서는 `view-billing`을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 AWS 계정의 도메인 등록 요금에 대한 결제 정보를 가져오려면

다음 `view-billing` 명령은 2018년 1월 1일(1514764800 Unix 시간)부터 2019년 12월 31일 자정(1577836800 Unix 시간)까지의 기간 동안 현재 계정에 대한 모든 도메인 관련 결제 레코드를 반환합니다.

이 명령은 `us-east-1` 리전에서만 실행됩니다. 기본 리전이 로 설정된 경우 `region` 파라미터를 생략할 `us-east-1`수 있습니다.

```
aws route53domains view-billing \
    --region us-east-1 \
    --start-time 1514764800 \
    --end-time 1577836800
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 Amazon Route 53 참조ViewBilling의 섹션을 참조하세요. Amazon Route 53 API

- 자세한 API 내용은 명령 참조ViewBilling의 섹션을 참조하세요. AWS CLI

## 를 사용하여 Route 53 Profiles 예제 AWS CLI

다음 코드 예제에서는 Route 53 프로파일과 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `associate-profile`

다음 코드 예시에서는 `associate-profile`을 사용하는 방법을 보여 줍니다.

AWS CLI

프로필을 연결하려면

다음 `associate-profile` 예제에서는 프로필을 에 연결합니다VPC.

```
aws route53profiles associate-profile \
    --name test-association \
    --profile-id rp-4987774726example \
    --resource-id vpc-0af3b96b3example
```

출력:

```
{
    "ProfileAssociation": {
        "CreationTime": 1710851336.527,
        "Id": "rpassoc-489ce212fexample",
        "ModificationTime": 1710851336.527,
        "Name": "test-association",
        "OwnerId": "123456789012",
        "ProfileId": "rp-4987774726example",
        "ResourceId": "vpc-0af3b96b3example",
        "Status": "CREATING",
        "StatusMessage": "Creating Profile Association"
    }
}
```

자세한 내용은 Amazon Route 53 개발자 안내서의 프로필 사용을 참조하세요.

- 자세한 API 내용은 명령 참조AssociateProfile의 섹션을 참조하세요. AWS CLI

### associate-resource-to-profile

다음 코드 예시에서는 associate-resource-to-profile을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스를 프로파일에 연결하려면

다음 associate-resource-to-profile 예제에서는 DNS 방화벽 규칙 그룹을 우선 순위 102와 프로파일에 연결합니다.

```
aws route53profiles associate-resource-to-profile \
    --name test-resource-association \
    --profile-id rp-4987774726example \
    --resource-arn arn:aws:route53resolver:us-east-1:123456789012:firewall-rule-
group/rslvr-frg-cfe7f72example \
    --resource-properties "{\"priority\": 102}"
```

출력:

```
{
    "ProfileResourceAssociation": {
```

```
        "CreationTime": 1710851216.613,
        "Id": "rpr-001913120a7example",
        "ModificationTime": 1710851216.613,
        "Name": "test-resource-association",
        "OwnerId": "123456789012",
        "ProfileId": "rp-4987774726example",
        "ResourceArn": "arn:aws:route53resolver:us-east-1:123456789012:firewall-
rule-group/rslvr-frg-cfe7f72example",
        "ResourceProperties": "{\"priority\":102}",
        "ResourceType": "FIREWALL_RULE_GROUP",
        "Status": "UPDATING",
        "StatusMessage": "Updating the Profile to DNS Firewall rule group
 association"
    }
}
```

- 자세한 API 내용은 명령 참조AssociateResourceToProfile의 섹션을 참조하세요. AWS CLI

### create-profile

다음 코드 예시에서는 create-profile을 사용하는 방법을 보여 줍니다.

AWS CLI

프로필을 생성하려면

다음 create-profile 예제에서는 프로필을 생성합니다.

```
aws route53profiles create-profile \
    --name test
```

출력:

```
{
    "Profile": {
        "Arn": "arn:aws:route53profiles:us-east-1:123456789012:profile/
rp-6ffe47d5example",
        "ClientToken": "2ca1a304-32b3-4f5f-bc4c-EXAMPLE11111",
        "CreationTime": 1710850903.578,
        "Id": "rp-6ffe47d5example",
        "ModificationTime": 1710850903.578,
        "Name": "test",
```

```
        "OwnerId": "123456789012",
        "ShareStatus": "NOT_SHARED",
        "Status": "COMPLETE",
        "StatusMessage": "Created Profile"
    }
}
```

- 자세한 API 내용은 명령 참조CreateProfile의 섹션을 참조하세요. AWS CLI

### `delete-profile`

다음 코드 예시에서는 `delete-profile`을 사용하는 방법을 보여 줍니다.

AWS CLI

프로필을 삭제하려면

다음 `delete-profile` 예제에서는 프로필을 삭제합니다.

```
aws route53profiles delete-profile \
    --profile-id rp-6ffe47d5example
```

출력:

```
{
    "Profile": {
        "Arn": "arn:aws:route53profiles:us-east-1:123456789012:profile/
rp-6ffe47d5example",
        "ClientToken": "0a15fec0-05d9-4f78-bec0-EXAMPLE11111",
        "CreationTime": 1710850903.578,
        "Id": "rp-6ffe47d5example",
        "ModificationTime": 1710850903.578,
        "Name": "test",
        "OwnerId": "123456789012",
        "ShareStatus": "NOT_SHARED",
        "Status": "DELETED",
        "StatusMessage": "Deleted Profile"
    }
}
```

- 자세한 API 내용은 명령 참조DeleteProfile의 섹션을 참조하세요. AWS CLI

### disassociate-profile

다음 코드 예시에서는 disassociate-profile을 사용하는 방법을 보여 줍니다.

AWS CLI

프로필 연결을 해제하려면

다음 disassociate-profile 예제에서는 에서 프로파일의 연결을 해제합니다VPC.

```
aws route53profiles disassociate-profile \
    --profile-id rp-4987774726example \
    --resource-id vpc-0af3b96b3example
```

출력:

```
{
    "ProfileAssociation": {
        "CreationTime": 1710851336.527,
        "Id": "rpassoc-489ce212fexample",
        "ModificationTime": 1710851401.362,
        "Name": "test-association",
        "OwnerId": "123456789012",
        "ProfileId": "rp-4987774726example",
        "ResourceId": "vpc-0af3b96b3example",
        "Status": "DELETING",
        "StatusMessage": "Deleting Profile Association"
    }
}
```

- 자세한 API 내용은 명령 참조DisassociateProfile의 섹션을 참조하세요. AWS CLI

### disassociate-resource-from-profile

다음 코드 예시에서는 disassociate-resource-from-profile을 사용하는 방법을 보여 줍니다.

AWS CLI

프로필에서 리소스 연결을 해제하려면

다음 disassociate-resource-from-profile 예제에서는 프로필에서 DNS 방화벽 규칙 그룹을 연결 해제합니다.

```
aws route53profiles disassociate-resource-from-profile \
    --profile-id rp-4987774726example \
    --resource-arn arn:aws:route53resolver:us-east-1:123456789012:firewall-rule-
group/rslvr-frg-cfe7f72example
```

출력:

```
{
    "ProfileResourceAssociation": {
        "CreationTime": 1710851216.613,
        "Id": "rpr-001913120a7example",
        "ModificationTime": 1710852624.36,
        "Name": "test-resource-association",
        "OwnerId": "123456789012",
        "ProfileId": "rp-4987774726example",
        "ResourceArn": "arn:aws:route53resolver:us-east-1:123456789012:firewall-
rule-group/rslvr-frg-cfe7f72example",
        "ResourceProperties": "{\"priority\":105}",
        "ResourceType": "FIREWALL_RULE_GROUP",
        "Status": "DELETING",
        "StatusMessage": "Deleting the Profile to DNS Firewall rule group
 association"
    }
}
```

- 자세한 API 내용은 명령 참조DisassociateResourceFromProfile의 섹션을 참조하세요. AWS CLI

### get-profile-association

다음 코드 예시에서는 get-profile-association을 사용하는 방법을 보여 줍니다.

AWS CLI

프로필 연결에 대한 정보를 가져오려면

다음은 지정된 프로필 연결에 대한 정보를 get-profile-association 반환합니다.

```
aws route53profiles get-profile-association \
    --profile-association-id rpassoc-489ce212fexample
```

출력:

```
{
    "ProfileAssociation": {
        "CreationTime": 1709338817.148,
        "Id": "rrpassoc-489ce212fexample",
        "ModificationTime": 1709338974.772,
        "Name": "test-association",
        "OwnerId": "123456789012",
        "ProfileId": "rp-4987774726example",
        "ResourceId": "vpc-0af3b96b3example",
        "Status": "COMPLETE",
        "StatusMessage": "Created Profile Association"
    }
}
```

- 자세한 API 내용은 명령 참조GetProfileAssociation의 섹션을 참조하세요. AWS CLI

### get-profile-resource-association

다음 코드 예시에서는 get-profile-resource-association을 사용하는 방법을 보여 줍니다.

AWS CLI

프로필과 연결된 리소스에 대한 정보를 가져오려면

다음은 지정된 리소스 연결에 대한 정보를 프로파일에 get-profile-resource-association 반환합니다.

```
aws route53profiles get-profile-resource-association \
    --profile-resource-association-id rpr-001913120a7example
```

출력:

```
{
    "ProfileResourceAssociation": {
        "CreationTime": 1710851216.613,
        "Id": "rpr-001913120a7example",
        "ModificationTime": 1710852303.798,
        "Name": "test-resource-association",
        "OwnerId": "123456789012",
        "ProfileId": "rp-4987774726example",
        "ResourceArn": "arn:aws:route53resolver:us-east-1:123456789012:firewall-
rule-group/rslvr-frg-cfe7f72example",
```

```
        "ResourceProperties": "{\"priority\":105}",
        "ResourceType": "FIREWALL_RULE_GROUP",
        "Status": "COMPLETE",
        "StatusMessage": "Completed creation of Profile to DNS Firewall rule group
 association"
    }
}
```

- 자세한 API 내용은 명령 참조GetProfileResourceAssociation의 섹션을 참조하세요. AWS CLI

### get-profile

다음 코드 예시에서는 get-profile을 사용하는 방법을 보여 줍니다.

AWS CLI

프로필에 대한 정보를 가져오려면

다음은 지정된 프로파일에 대한 정보를 get-profile 반환합니다.

```
aws route53profiles get-profile \
    --profile-id rp-4987774726example
```

출력:

```
{
    "Profile": {
        "Arn": "arn:aws:route53profiles:us-east-1:123456789012:profile/
rp-4987774726example",
        "ClientToken": "0cbc5ae7-4921-4204-bea9-EXAMPLE11111",
        "CreationTime": 1710851044.288,
        "Id": "rp-4987774726example",
        "ModificationTime": 1710851044.288,
        "Name": "test",
        "OwnerId": "123456789012",
        "ShareStatus": "NOT_SHARED",
        "Status": "COMPLETE",
        "StatusMessage": "Created Profile"
    }
}
```

- 자세한 API 내용은 명령 참조GetProfile의 섹션을 참조하세요. AWS CLI

### list-profile-associations

다음 코드 예시에서는 list-profile-associations을 사용하는 방법을 보여 줍니다.

AWS CLI

프로필 연결을 나열하려면

다음은 AWS 계정의 프로필 연결을 list-profile-associations 나열합니다.

```
aws route53profiles list-profile-associations
```

출력:

```
{
    "ProfileAssociations": [
        {
            "CreationTime": 1709338817.148,
            "Id": "rpassoc-489ce212fexample",
            "ModificationTime": 1709338974.772,
            "Name": "test-association",
            "OwnerId": "123456789012",
            "ProfileId": "rp-4987774726example",
            "ResourceId": "vpc-0af3b96b3example",
            "Status": "COMPLETE",
            "StatusMessage": "Created Profile Association"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListProfileAssociations의 섹션을 참조하세요. AWS CLI

### list-profile-resource-associations

다음 코드 예시에서는 list-profile-resource-associations을 사용하는 방법을 보여 줍니다.

AWS CLI

프로필 리소스 연결을 나열하려면

다음은 지정된 프로파일에 대한 프로파일 리소스 연결을 list-profile-resource-associations 나열합니다.

```
aws route53profiles list-profile-resource-associations \
    --profile-id rp-4987774726example
```

출력:

```
{
     "ProfileResourceAssociations": [
         {
             "CreationTime": 1710851216.613,
             "Id": "rpr-001913120a7example",
             "ModificationTime": 1710851216.613,
             "Name": "test-resource-association",
             "OwnerId": "123456789012",
             "ProfileId": "rp-4987774726example",
             "ResourceArn": "arn:aws:route53resolver:us-
east-1:123456789012:firewall-rule-group/rslvr-frg-cfe7f72example",
             "ResourceProperties": "{\"priority\":102}",
             "ResourceType": "FIREWALL_RULE_GROUP",
             "Status": "COMPLETE",
             "StatusMessage": "Completed creation of Profile to DNS Firewall rule
 group association"
         }
     ]
 }
```

- 자세한 API 내용은 명령 참조ListProfileResourceAssociations의 섹션을 참조하세요. AWS CLI

### `list-profiles`

다음 코드 예시에서는 `list-profiles`을 사용하는 방법을 보여 줍니다.

AWS CLI

프로필을 나열하려면

다음은 AWS 계정의 프로필을 `list-profiles` 나열하고 이에 대한 추가 정보를 표시합니다.

```
aws route53profiles list-profiles
```

출력:

```
{
```

```
    "ProfileSummaries": [
        {
            "Arn": "arn:aws:route53profiles:us-east-1:123456789012:profile/
rp-4987774726example",
            "Id": "rp-4987774726example",
            "Name": "test",
            "ShareStatus": "NOT_SHARED"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListProfiles의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스의 태그를 나열하려면

다음은 지정된 리소스에 대한 태그를 `list-tags-for-resource` 나열합니다.

```
aws route53profiles list-tags-for-resource \
    --resource-arn arn:aws:route53profiles:us-east-1:123456789012:profile/
rp-4987774726example
```

출력:

```
{
    "Tags": {
        "my-key-2": "my-value-2",
        "my-key-1": "my-value-1"
    }
}
```

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### `update-profile-resource-association`

다음 코드 예시에서는 `update-profile-resource-association`을 사용하는 방법을 보여 줍니다.

AWS CLI

프로필에 연결된 리소스를 업데이트하려면

다음은 프로필에 연결된 DNS 방화벽 규칙 그룹의 우선 순위를 update-profile-resource-association 업데이트합니다.

```
aws route53profiles update-profile-resource-association \
    --profile-resource-association-id rpr-001913120a7example \
    --resource-properties "{\"priority\": 105}"
```

출력:

```
{
    "ProfileResourceAssociation": {
        "CreationTime": 1710851216.613,
        "Id": "rpr-001913120a7example",
        "ModificationTime": 1710852303.798,
        "Name": "test-resource-association",
        "OwnerId": "123456789012",
        "ProfileId": "rp-4987774726example",
        "ResourceArn": "arn:aws:route53resolver:us-east-1:123456789012:firewall-
rule-group/rslvr-frg-cfe7f72example",
        "ResourceProperties": "{\"priority\":105}",
        "ResourceType": "FIREWALL_RULE_GROUP",
        "Status": "UPDATING",
        "StatusMessage": "Updating the Profile to DNS Firewall rule group
 association"
    }
}
```

- 자세한 API 내용은 명령 참조UpdateProfileResourceAssociation의 섹션을 참조하세요. AWS CLI

## 를 사용하여 Route 53 Resolver 예제 AWS CLI

다음 코드 예제에서는 Route 53 Resolver와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `associate-firewall-rule-group`

다음 코드 예시에서는 `associate-firewall-rule-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

방화벽 규칙 그룹을 에 연결하려면 VPC

다음 `associate-firewall-rule-group` 예제에서는 DNS 방화벽 규칙 그룹을 Amazon 에 연결합니다VPC.

```
aws route53resolver associate-firewall-rule-group \
    --name test-association \
    --firewall-rule-group-id rslvr-frg-47f93271fexample \
    --vpc-id vpc-31e92222 \
    --priority 101
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon [Route 53 개발자 안내서의 VPC 및 Route 53 Resolver DNS Firewall 규칙 그룹 간의 연결 관리를 참조하세요](). Amazon Route 53

- 자세한 API 내용은 명령 참조[AssociateFirewallRuleGroup]()의 섹션을 참조하세요. AWS CLI

### `associate-resolver-endpoint-ip-address`

다음 코드 예시에서는 `associate-resolver-endpoint-ip-address`을 사용하는 방법을 보여 줍니다.

AWS CLI

다른 IP 주소를 Resolver 엔드포인트에 연결하려면

다음 `associate-resolver-endpoint-ip-address` 예제에서는 다른 IP 주소를 인바운드 Resolver 엔드포인트와 연결합니다. 서브넷 ID만 지정하고 `--ip-address` 파라미터에서 IP 주소를 생략하면 Resolver는 지정된 서브넷의 사용 가능한 IP 주소 중에서 IP 주소를 선택합니다.

```
aws route53resolver associate-resolver-endpoint-ip-address \
    --resolver-endpoint-id rslvr-in-497098ad5example \
    --ip-address="SubnetId=subnet-12d8exam,Ip=192.0.2.118"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Route 53 개발자 안내서의 인바운드 엔드포인트를 생성하거나 편집할 때 지정하는 값을 참조하세요.

- 자세한 API 내용은 명령 참조AssociateResolverEndpointIpAddress의 섹션을 참조하세요. AWS CLI

### associate-resolver-rule

다음 코드 예시에서는 associate-resolver-rule을 사용하는 방법을 보여 줍니다.

AWS CLI

Resolver 규칙을 와 연결하려면 VPC

다음 associate-resolver-rule 예제에서는 Resolver 규칙을 Amazon 와 연결합니다VPC. 명령을 실행한 후 Resolver는 전달된 DNS 쿼리의 도메인 이름과 같은 규칙의 설정을 기반으로 네트워크에 쿼리를 전달하기 시작합니다.

```
aws route53resolver associate-resolver-rule \
    --name my-resolver-rule-association \
    --resolver-rule-id rslvr-rr-42b60677c0example \
    --vpc-id vpc-304bexam
```

출력:

```
{
    "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."
    }
```

```
}
```

자세한 내용은 Amazon Route 53 개발자 안내서의 네트워크에 아웃바운드 DNS 쿼리 전달을 참조하세요.

- 자세한 API 내용은 명령 참조AssociateResolverRule의 섹션을 참조하세요. AWS CLI

### `create-firewall-domain-list`

다음 코드 예시에서는 `create-firewall-domain-list`을 사용하는 방법을 보여 줍니다.

AWS CLI

Route 53 Resolver DNS Firewall 도메인 목록을 생성하려면

다음 `create-firewall-domain-list` 예제에서는 AWS 계정에 테스트라는 Route 53 Resolver DNS Firewall 도메인 목록을 생성합니다.

```
aws route53resolver create-firewall-domain-list \
    --creator-request-id my-request-id \
    --name test
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Route 53 개발자 안내서의 자체 도메인 목록 관리를 참조하세요.

- 자세한 API 내용은 명령 참조CreateFirewallDomainList의 섹션을 참조하세요. AWS CLI

### create-firewall-rule-group

다음 코드 예시에서는 create-firewall-rule-group을 사용하는 방법을 보여 줍니다.

AWS CLI

방화벽 규칙 그룹을 생성하려면

다음 create-firewall-rule-group 예제에서는 DNS 방화벽 규칙 그룹을 생성합니다.

```
aws route53resolver create-firewall-rule-group \
    --creator-request-id my-request-id \
    --name test
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Route 53 개발자 안내서의 DNS 방화벽에서 규칙 그룹 및 규칙 관리를 참조하세요.

- 자세한 API 내용은 명령 참조CreateFirewallRuleGroup의 섹션을 참조하세요. AWS CLI

### create-firewall-rule

다음 코드 예시에서는 create-firewall-rule을 사용하는 방법을 보여 줍니다.

AWS CLI

방화벽 규칙을 생성하려면

다음 `create-firewall-rule` 예제에서는 방화벽 도메인 목록에 나열된 도메인에 대한 DNS 방화벽 규칙에 DNS 방화벽 규칙을 생성합니다.

```
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
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Route 53 개발자 안내서의 DNS 방화벽에서 규칙 그룹 및 규칙 관리를 참조하세요.

- 자세한 API 내용은 명령 참조CreateFirewallRule의 섹션을 참조하세요. AWS CLI

### `create-resolver-endpoint`

다음 코드 예시에서는 `create-resolver-endpoint`을 사용하는 방법을 보여 줍니다.

AWS CLI

인바운드 Resolver 엔드포인트를 생성하려면

다음 `create-resolver-endpoint` 예제에서는 인바운드 Resolver 엔드포인트를 생성합니다. 동일한 명령을 사용하여 인바운드 및 아웃바운드 엔드포인트를 모두 생성할 수 있습니다.

aws route53resolver create-resolver-endpoint --name my-inbound-endpoint --creator-request-id 2020-01-01-18:47 --security-group-ids 'sg-f62bexam' --direction INBOUND --ip-addresses SubnetId=subnet-ba47exam,Ip=192.0.2.255 SubnetId=subnet-12d8exam,Ip=192.0.2.254

출력:

```
{
    "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"
    }
}
```

아웃바운드 Resolver 엔드포인트를 생성하려면

다음 `create-resolver-endpoint` 예제에서는 JSON형식이 지정된 문서 의 값을 사용하여 아웃바운드 해석기 엔드포인트를 생성합니다`create-outbound-resolver-endpoint.json`.

```
aws route53resolver create-resolver-endpoint \
    --cli-input-json file://c:\temp\create-outbound-resolver-endpoint.json
```

`create-outbound-resolver-endpoint.json`의 콘텐츠:

```
{
```

```
    "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"
       }
    ]
}
```

자세한 내용은 Amazon Route 53 개발자 안내서의 VPCs 와 네트워크 간의 DNS 쿼리 해결을 참조하세요.

- 자세한 API 내용은 명령 참조CreateResolverEndpoint의 섹션을 참조하세요. AWS CLI

### `create-resolver-rule`

다음 코드 예시에서는 `create-resolver-rule`을 사용하는 방법을 보여 줍니다.

AWS CLI

해석기 규칙을 생성하려면

다음 `create-resolver-rule` 예제에서는 Resolver 전달 규칙을 생성합니다. 이 규칙은 아웃바운드 엔드포인트 rslvr-out-d5e5920e37example을 사용하여 에 대한 DNS 쿼리`example.com`를 IP 주소 10.24.8.75 및 10.24.8.156으로 전달합니다.

```
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"
```

출력:

```
{
    "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."
    }
}
```

- 자세한 API 내용은 명령 참조CreateResolverRule의 섹션을 참조하세요. AWS CLI

### delete-firewall-domain-list

다음 코드 예시에서는 delete-firewall-domain-list을 사용하는 방법을 보여 줍니다.

AWS CLI

Route 53 Resolver DNS 방화벽 도메인 목록을 삭제하려면

다음 `delete-firewall-domain-list` 예제에서는 AWS 계정에서 테스트라는 Route 53 Resolver DNS Firewall 도메인 목록을 삭제합니다.

```
aws route53resolver delete-firewall-domain-list \
    --firewall-domain-list-id rslvr-fdl-9e956e9ffexample
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Route 53 개발자 안내서의 자체 도메인 목록 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteFirewallDomainList의 섹션을 참조하세요. AWS CLI

### `delete-firewall-rule-group`

다음 코드 예시에서는 `delete-firewall-rule-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

방화벽 규칙 그룹을 삭제하려면

다음 `delete-firewall-rule-group` 예제에서는 방화벽 규칙 그룹을 삭제합니다.

```
aws route53resolver delete-firewall-rule-group \
```

```
    --firewall-rule-group-id rslvr-frg-47f93271fexample
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Route 53 개발자 안내서의 DNS 방화벽에서 규칙 그룹 및 규칙 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteFirewallRuleGroup의 섹션을 참조하세요. AWS CLI

### delete-firewall-rule

다음 코드 예시에서는 delete-firewall-rule을 사용하는 방법을 보여 줍니다.

AWS CLI

방화벽 규칙을 삭제하려면

다음 delete-firewall-rule 예제에서는 지정된 방화벽 규칙을 삭제합니다.

```
aws route53resolver delete-firewall-rule \
    --firewall-rule-group-id rslvr-frg-47f93271fexample \
    --firewall-domain-list-id rslvr-fdl-9e956e9ffexample
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Route 53 개발자 안내서의 DNS 방화벽에서 규칙 그룹 및 규칙 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteFirewallRule의 섹션을 참조하세요. AWS CLI

### `delete-resolver-endpoint`

다음 코드 예시에서는 `delete-resolver-endpoint`을 사용하는 방법을 보여 줍니다.

AWS CLI

Resolver 엔드포인트를 삭제하려면

다음 `delete-resolver-endpoint` 예제에서는 지정된 엔드포인트를 삭제합니다.

중요 인바운드 엔드포인트를 삭제하면 엔드포인트에 VPC 지정한 에서 네트워크의 DNS 쿼리가 더 이상 Resolver로 전달되지 않습니다. 아웃바운드 엔드포인트를 삭제하면 Resolver는 삭제된 아웃바운드 엔드포인트를 지정하는 규칙에 대해 에서 네트워크VPC로 DNS 쿼리 전달을 중지합니다.

```
aws route53resolver delete-resolver-endpoint \
    --resolver-endpoint-id rslvr-in-497098ad59example
```

출력:

```
{
    "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"
    }
}
```

- 자세한 API 내용은 명령 참조DeleteResolverEndpoint의 섹션을 참조하세요. AWS CLI

### `delete-resolver-rule`

다음 코드 예시에서는 `delete-resolver-rule`을 사용하는 방법을 보여 줍니다.

AWS CLI

해석기 규칙을 삭제하려면

다음 `delete-resolver-rule` 예제에서는 지정된 규칙을 삭제합니다.

참고 규칙이 와 연결된 경우 먼저 에서 규칙을 연결 해제VPCs해야 규칙을 삭제할 VPCs 수 있습니다.

```
aws route53resolver delete-resolver-rule \
    --resolver-rule-id rslvr-rr-5b3809426bexample
```

출력:

```
{
    "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"
    }
}
```

- 자세한 API 내용은 명령 참조DeleteResolverRule의 섹션을 참조하세요. AWS CLI

### `disassociate-firewall-rule-group`

다음 코드 예시에서는 `disassociate-firewall-rule-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

방화벽 규칙 그룹을 에서 연결 해제하려면 VPC

다음 `disassociate-firewall-rule-group` 예제에서는 Amazon 에서 DNS 방화벽 규칙 그룹을 연결 해제합니다VPC.

```
aws route53resolver disassociate-firewall-rule-group \
    --firewall-rule-group-association-id rslvr-frgassoc-57e8873d7example
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Route 53 개발자 안내서의 VPC 및 Route 53 Resolver DNS Firewall 규칙 그룹 간의 연결 관리를 참조하세요. Amazon Route 53

- 자세한 API 내용은 명령 참조DisassociateFirewallRuleGroup의 섹션을 참조하세요. AWS CLI

### `disassociate-resolver-endpoint-ip-address`

다음 코드 예시에서는 `disassociate-resolver-endpoint-ip-address`을 사용하는 방법을 보여 줍니다.

AWS CLI

Resolver 엔드포인트에서 IP 주소 연결을 해제하려면

다음 `disassociate-resolver-endpoint-ip-address` 예제에서는 지정된 Resolver 인바운드 또는 아웃바운드 엔드포인트에서 IP 주소를 제거합니다.

참고 엔드포인트에는 IP 주소가 두 개 이상 있어야 합니다. 엔드포인트에 현재 두 개의 IP 주소만 있고 한 주소를 다른 주소로 바꾸려면 먼저 associate-resolver-endpoint-ip-address를 사용하여 새 IP 주소를 연결해야 합니다. 그런 다음 엔드포인트에서 원래 IP 주소 중 하나를 연결 해제할 수 있습니다.

```
aws route53resolver disassociate-resolver-endpoint-ip-address \
    --resolver-endpoint-id rslvr-in-f9ab8a03f1example \
    --ip-address="SubnetId=subnet-12d8a459,Ip=172.31.40.121"
```

출력:

```
{
    "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"
    }
}
```

- 자세한 API 내용은 명령 참조DisassociateResolverEndpointIpAddress의 섹션을 참조하세요. AWS CLI

### `disassociate-resolver-rule`

다음 코드 예시에서는 `disassociate-resolver-rule`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon에서 Resolver 규칙을 연결 해제하려면 VPC

다음 `disassociate-resolver-rule` 예제에서는 지정된 해석기 규칙과 지정된 간의 연결을 제거합니다VPC. 다음과 같은 경우 에서 규칙을 연결 해제VPC할 수 있습니다.

이 에서 시작된 DNS 쿼리의 경우 Resolver가 규칙에 지정된 도메인 이름에 대한 쿼리를 네트워크에 전달하는 것을 중지하도록 VPC해야 합니다. 전달 규칙을 삭제하려고 합니다. 규칙이 현재 하나 이상의 와 연결되어 있는 경우 VPCs규칙을 삭제하기 VPCs 전에 모든 에서 규칙을 연결 해제해야 합니다.

```
aws route53resolver disassociate-resolver-rule \
    --resolver-rule-id rslvr-rr-4955cb98ceexample \
```

```
    --vpc-id vpc-304bexam
```

출력:

```
{
    "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"
    }
}
```

- 자세한 API 내용은 명령 참조DisassociateResolverRule의 섹션을 참조하세요. AWS CLI

### `get-firewall-config`

다음 코드 예시에서는 `get-firewall-config`을 사용하는 방법을 보여 줍니다.

AWS CLI

에 대한 방화벽 구성을 가져오려면 VPC

다음 `get-firewall-config` 예제에서는 지정된 에 대한 DNS 방화벽 동작을 검색합니다VPC.

```
aws route53resolver get-firewall-config \
    --resource-id vpc-31e92222
```

출력:

```
{
    "FirewallConfig": {
        "Id": "rslvr-fc-86016850cexample",
        "ResourceId": "vpc-31e9222",
        "OwnerId": "123456789012",
        "FirewallFailOpen": "DISABLED"
    }
}
```

자세한 내용은 Amazon Route 53 개발자 안내서의 DNS 방화벽 VPC 구성을 참조하세요.

- 자세한 API 내용은 명령 참조GetFirewallConfig의 섹션을 참조하세요. AWS CLI

### get-firewall-domain-list

다음 코드 예시에서는 get-firewall-domain-list을 사용하는 방법을 보여 줍니다.

AWS CLI

Route 53 Resolver DNS 방화벽 도메인 목록을 가져오려면

다음 get-firewall-domain-list 예제에서는 지정한 ID로 도메인 목록을 검색합니다.

```
aws route53resolver get-firewall-domain-list \
    --firewall-domain-list-id rslvr-fdl-42b60677cexample
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Route 53 개발자 안내서의 자체 도메인 목록 관리를 참조하세요.

- 자세한 API 내용은 명령 참조GetFirewallDomainList의 섹션을 참조하세요. AWS CLI

### get-firewall-rule-group-association

다음 코드 예시에서는 get-firewall-rule-group-association을 사용하는 방법을 보여 줍니다.

AWS CLI

방화벽 규칙 그룹 연결을 가져오려면

다음 `get-firewall-rule-group-association` 예제에서는 방화벽 규칙 그룹 연결을 검색합니다.

```
aws route53resolver get-firewall-rule-group-association \
    --firewall-rule-group-association-id rslvr-frgassoc-57e8873d7example
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Route 53 개발자 안내서의 VPC 및 Route 53 Resolver DNS Firewall 규칙 그룹 간의 연결 관리를 참조하세요.

- 자세한 API 내용은 명령 참조GetFirewallRuleGroupAssociation의 섹션을 참조하세요. AWS CLI

### `get-firewall-rule-group-policy`

다음 코드 예시에서는 `get-firewall-rule-group-policy`을 사용하는 방법을 보여 줍니다.

AWS CLI

정책을 가져오 AWS IAM려면

다음 get-firewall-rule-group-policy 예제에서는 지정된 규칙 그룹을 공유하기 위한 AWS 자격 증명 및 액세스 관리(AWS IAM) 정책을 가져옵니다.

```
aws route53resolver get-firewall-rule-group-policy \
    --arn arn:aws:route53resolver:us-west-2:AWS_ACCOUNT_ID:firewall-rule-group/
rslvr-frg-47f93271fexample
```

출력:

```
{
    "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\"}]}"
}
```

자세한 내용은 Amazon Route 53 개발자 안내서의 DNS 방화벽에서 규칙 그룹 및 규칙 관리를 참조하세요.

- 자세한 API 내용은 명령 참조GetFirewallRuleGroupPolicy의 섹션을 참조하세요. AWS CLI

### get-firewall-rule-group

다음 코드 예시에서는 get-firewall-rule-group을 사용하는 방법을 보여 줍니다.

AWS CLI

방화벽 규칙 그룹을 가져오려면

다음 get-firewall-rule-group 예제에서는 사용자가 제공한 ID가 있는 DNS 방화벽 규칙 그룹에 대한 정보를 검색합니다.

```
aws route53resolver get-firewall-rule-group \
    --firewall-rule-group-id rslvr-frg-47f93271fexample
```

출력:

```
{
```

```
    "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"
    }
}
```

자세한 내용은 Amazon Route 53 개발자 안내서의 DNS 방화벽에서 규칙 그룹 및 규칙 관리를 참조하세요.

- 자세한 API 내용은 명령 참조GetFirewallRuleGroup의 섹션을 참조하세요. AWS CLI

### get-resolver-endpoint

다음 코드 예시에서는 get-resolver-endpoint을 사용하는 방법을 보여 줍니다.

AWS CLI

Resolver 엔드포인트에 대한 정보를 가져오려면

다음 get-resolver-endpoint 예제에서는 아웃바운드 지정 엔드포인트에 대한 세부 정보를 표시합니다. 해당 엔드포인트 ID를 지정하여 인바운드 및 아웃바운드 엔드포인트 모두에 get-resolver-endpoint 를 사용할 수 있습니다.

```
aws route53resolver get-resolver-endpoint \
    --resolver-endpoint-id rslvr-out-d5e5920e37example
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Route 53 개발자 안내서의 인바운드 엔드포인트를 생성하거나 편집할 때 지정하는 값을 참조하세요.

- 자세한 API 내용은 명령 참조GetResolverEndpoint의 섹션을 참조하세요. AWS CLI

### get-resolver-rule-association

다음 코드 예시에서는 get-resolver-rule-association을 사용하는 방법을 보여 줍니다.

AWS CLI

Resolver 규칙과 간의 연결에 대한 정보를 가져오려면 VPC

다음 get-resolver-rule-association 예제에서는 지정된 Resolver 규칙과 간의 연결에 대한 세부 정보를 표시합니다VPC. 를 VPC 사용하여 해석기 규칙과 를 연결합니다associate-resolver-rule.

```
aws route53resolver get-resolver-rule-association \
    --resolver-rule-association-id rslvr-rrassoc-d61cbb2c8bexample
```

출력:

```
{
    "ResolverRuleAssociation": {
        "Id": "rslvr-rrassoc-d61cbb2c8bexample",
        "ResolverRuleId": "rslvr-rr-42b60677c0example",
```

```
        "Name": "my-resolver-rule-association",
        "VPCId": "vpc-304bexam",
        "Status": "COMPLETE",
        "StatusMessage": ""
    }
}
```

- 자세한 API 내용은 명령 참조GetResolverRuleAssociation의 섹션을 참조하세요. AWS CLI

### get-resolver-rule

다음 코드 예시에서는 get-resolver-rule을 사용하는 방법을 보여 줍니다.

AWS CLI

Resolver 규칙에 대한 정보를 가져오려면

다음 get-resolver-rule 예제에서는 규칙이 DNS 쿼리를 전달하는 도메인 이름 및 규칙과 연결된 아웃바운드 해석기 엔드포인트의 ID와 같은 지정된 해석기 규칙에 대한 세부 정보를 표시합니다.

```
aws route53resolver get-resolver-rule \
    --resolver-rule-id rslvr-rr-42b60677c0example
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Route 53 개발자 안내서의 규칙을 생성하거나 편집할 때 지정하는 값을 참조하세요.

- 자세한 API 내용은 명령 참조GetResolverRule의 섹션을 참조하세요. AWS CLI

### `import-firewall-domains`

다음 코드 예시에서는 `import-firewall-domains`을 사용하는 방법을 보여 줍니다.

AWS CLI

도메인을 도메인 목록으로 가져오려면

다음 `import-firewall-domains` 예제에서는 파일에서 지정한 DNS 방화벽 도메인 목록으로 도메인 세트를 가져옵니다.

```
aws route53resolver import-firewall-domains \
    --firewall-domain-list-id rslvr-fdl-d61cbb2cbexample \
    --operation REPLACE \
    --domain-file-url s3://PATH/TO/YOUR/FILE
```

출력:

```
{
    "Id": "rslvr-fdl-d61cbb2cbexample",
    "Name": "test",
    "Status": "IMPORTING",
    "StatusMessage": "Importing domains from provided file."
}
```

자세한 내용은 Amazon Route 53 개발자 안내서의 자체 도메인 목록 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ImportFirewallDomains의 섹션을 참조하세요. AWS CLI

### list-firewall-configs

다음 코드 예시에서는 list-firewall-configs을 사용하는 방법을 보여 줍니다.

AWS CLI

방화벽 구성을 나열하려면

다음 list-firewall-configs 예제에서는 DNS 방화벽 구성을 나열합니다.

```
aws route53resolver list-firewall-configs
```

출력:

```
{
    "FirewallConfigs": [
        {
            "Id": "rslvr-fc-86016850cexample",
            "ResourceId": "vpc-31e92222",
            "OwnerId": "123456789012",
            "FirewallFailOpen": "DISABLED"
        }
    ]
}
```

자세한 내용은 Amazon Route 53 개발자 안내서의 DNS 방화벽 VPC 구성을 참조하세요.

- 자세한 API 내용은 명령 참조ListFirewallConfigs의 섹션을 참조하세요. AWS CLI

### list-firewall-domain-lists

다음 코드 예시에서는 list-firewall-domain-lists을 사용하는 방법을 보여 줍니다.

AWS CLI

Route 53 Resolver DNS Firewall 도메인 목록을 모두 나열하려면

다음 list-firewall-domain-lists 예제에서는 모든 도메인 목록을 나열합니다.

```
aws route53resolver list-firewall-domain-lists
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Route 53 개발자 안내서의 Route 53 Resolver DNS Firewall 도메인 목록을 참조하세요. Amazon Route 53

- 자세한 API 내용은 명령 참조ListFirewallDomainLists의 섹션을 참조하세요. AWS CLI

### `list-firewall-domains`

다음 코드 예시에서는 `list-firewall-domains`을 사용하는 방법을 보여 줍니다.

AWS CLI

도메인 목록에 도메인을 나열하려면

다음 `list-firewall-domains` 예제에서는 지정한 DNS 방화벽 도메인 목록의 도메인을 나열합니다.

```
aws route53resolver list-firewall-domains \
    --firewall-domain-list-id rslvr-fdl-d61cbb2cbexample
```

출력:

```
{
    "Domains": [
        "test1.com.",
        "test2.com.",
        "test3.com."
    ]
}
```

자세한 내용은 Amazon Route 53 개발자 안내서의 자체 도메인 목록 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ListFirewallDomains의 섹션을 참조하세요. AWS CLI

### `list-firewall-rule-group-associations`

다음 코드 예시에서는 `list-firewall-rule-group-associations`을 사용하는 방법을 보여 줍니다.

AWS CLI

DNS 방화벽 규칙 그룹 연결을 나열하려면

다음 `list-firewall-rule-group-associations` 예제에서는 Amazon 와의 DNS 방화벽 규칙 그룹 연결을 나열합니다VPCs.

```
aws route53resolver list-firewall-rule-group-associations
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Route 53 개발자 안내서의 VPC 및 Route 53 Resolver DNS Firewall 규칙 그룹 간의 연결 관리를 참조하세요. Amazon Route 53

- 자세한 API 내용은 명령 참조ListFirewallRuleGroupAssociations의 섹션을 참조하세요. AWS CLI

### `list-firewall-rule-groups`

다음 코드 예시에서는 `list-firewall-rule-groups`을 사용하는 방법을 보여 줍니다.

AWS CLI

방화벽 규칙 그룹 목록을 가져오려면

다음 `list-firewall-rule-groups` 예제에서는 DNS 방화벽 규칙 그룹을 나열합니다.

```
aws route53resolver list-firewall-rule-groups
```

출력:

```
{
    "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"
        }
```

```
    ]
}
```

자세한 내용은 Amazon Route 53 개발자 안내서의 DNS 방화벽에서 규칙 그룹 및 규칙 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ListFirewallRuleGroups의 섹션을 참조하세요. AWS CLI

### `list-firewall-rules`

다음 코드 예시에서는 `list-firewall-rules`을 사용하는 방법을 보여 줍니다.

AWS CLI

방화벽 규칙을 나열하려면

다음 `list-firewall-rules` 예제에서는 DNS 방화벽 규칙 그룹 내의 모든 방화벽 규칙을 나열합니다.

```
aws route53resolver list-firewall-rules \
    --firewall-rule-group-id rslvr-frg-47f93271fexample
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Route 53 개발자 안내서의 DNS 방화벽에서 규칙 그룹 및 규칙 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ListFirewallRules의 섹션을 참조하세요. AWS CLI

### `list-resolver-endpoint-ip-addresses`

다음 코드 예시에서는 `list-resolver-endpoint-ip-addresses`을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 인바운드 또는 아웃바운드 엔드포인트의 IP 주소를 나열하려면

다음 `list-resolver-endpoint-ip-addresses` 예제에서는 인바운드 엔드포인트 와 연결된 IP 주소에 대한 정보를 나열합니다`rslvr-in-f9ab8a03f1example`. 해당 엔드포인트 ID를 지정하여 아웃바운드 엔드포인트`list-resolver-endpoint-ip-addresses`에 를 사용할 수도 있습니다.

```
aws route53resolver list-resolver-endpoint-ip-addresses \
    --resolver-endpoint-id rslvr-in-f9ab8a03f1example
```

출력:

```
{
    "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"
```

```
        }
    ]
}
```

출력의 값에 대한 자세한 내용은 Amazon Route 53 개발자 안내서의 인바운드 엔드포인트 생성 또는 편집 시 지정하는 값 및 아웃바운드 엔드포인트 생성 또는 편집 시 지정하는 값을 참조하세요.

- 자세한 API 내용은 명령 참조ListResolverEndpointIpAddresses의 섹션을 참조하세요. AWS CLI

### `list-resolver-endpoints`

다음 코드 예시에서는 `list-resolver-endpoints`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 리전에서 Resolver 엔드포인트를 나열하려면

다음 `list-resolver-endpoints` 예제에서는 현재 계정에 있는 인바운드 및 아웃바운드 Resolver 엔드포인트를 나열합니다.

```
aws route53resolver list-resolver-endpoints
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListResolverEndpoints의 섹션을 참조하세요. AWS CLI

### `list-resolver-rule-associations`

다음 코드 예시에서는 `list-resolver-rule-associations`을 사용하는 방법을 보여 줍니다.

AWS CLI

Resolver 규칙과 간의 연결을 나열하려면 VPCs

다음 `list-resolver-rule-associations` 예제에서는 해석기 규칙과 현재 AWS 계정 간의 연결을 나열VPCs합니다.

```
aws route53resolver list-resolver-rule-associations
```

출력:

```
{
    "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": ""
        }
    ]
}
```

자세한 내용은 Amazon Route 53 개발자 안내서의 Route 53 Resolver가 에서 네트워크VPCs로 DNS 쿼리를 전달하는 방법을 참조하세요. Amazon Route 53

- 자세한 API 내용은 명령 참조ListResolverRuleAssociations의 섹션을 참조하세요. AWS CLI

### `list-resolver-rules`

다음 코드 예시에서는 `list-resolver-rules`을 사용하는 방법을 보여 줍니다.

AWS CLI

해석기 규칙을 나열하려면

다음 `list-resolver-rules` 예제에서는 현재 AWS 계정의 모든 Resolver 규칙을 나열합니다.

```
aws route53resolver list-resolver-rules
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Route 53 개발자 안내서의 Route 53 Resolver가 에서 네트워크VPCs로 DNS 쿼리를 전달하는 방법을 참조하세요. Amazon Route 53

- 자세한 API 내용은 명령 참조ListResolverRules의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

Resolver 리소스의 태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 지정된 Resolver 규칙에 할당된 태그를 나열합니다.

```
aws route53resolver list-tags-for-resource \
    --resource-arn "arn:aws:route53resolver:us-west-2:111122223333:resolver-rule/rslvr-rr-42b60677c0example"
```

출력:

```
{
    "Tags": [
        {
            "Key": "my-key-1",
            "Value": "my-value-1"
        },
        {
            "Key": "my-key-2",
            "Value": "my-value-2"
        }
    ]
}
```

비용 할당에 태그를 사용하는 방법에 대한 자세한 내용은 AWS 결제 및 비용 관리 사용 설명서의 비용 할당 태그 사용을 참조하세요.

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### `put-firewall-rule-group-policy`

다음 코드 예시에서는 `put-firewall-rule-group-policy`을 사용하는 방법을 보여 줍니다.

AWS CLI

방화벽 규칙 그룹 정책을 공유할 정책을 연결 AWS IAM하려면

다음 `put-firewall-rule-group-policy` 예제에서는 규칙 그룹을 공유하기 위한 AWS 자격 증명 및 액세스 관리(AWS IAM) 정책을 연결합니다.

```
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\"}]}"
```

출력:

```
{
    "ReturnValue": true
}
```

자세한 내용은 Amazon Route 53 개발자 안내서의 DNS 방화벽에서 규칙 그룹 및 규칙 관리를 참조하세요.

- 자세한 API 내용은 명령 참조PutFirewallRuleGroupPolicy의 섹션을 참조하세요. AWS CLI

### put-resolver-rule-policy

다음 코드 예시에서는 put-resolver-rule-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

해석기 규칙을 다른 AWS 계정과 공유하려면

다음 put-resolver-rule-policy 예제에서는 다른 AWS 계정과 공유하려는 Resolver 규칙, 규칙을 공유하려는 계정, 계정에서 규칙에 대해 수행할 수 있게 하려는 규칙 관련 작업을 지정합니다.

참고 규칙을 생성한 것과 동일한 계정의 보안 인증 정보를 사용하여 이 명령을 실행해야 합니다.

```
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\" ] } ] }"
```

출력:

```
{
    "ReturnValue": true
}
```

를 실행한 후 다음 두 개의 Resource Access Manager(RAM) 명령을 실행할 `put-resolver-rule-policy`수 있습니다. 규칙을 공유하려는 계정을 사용해야 합니다.

`get-resource-share-invitations` 는 값 를 반환합니다`resourceShareInvitationArn`. 공유 규칙을 사용하도록 초대를 수락하려면 이 값이 필요합니다.`accept-resource-share-invitation` 는 공유 규칙을 사용하도록 초대를 수락합니다.

자세한 내용은 다음 설명서를 참조하세요.

Amazon Route 53 개발자 안내서의 get-resource-share-invitationsaccept-resource-share-invitations다른 AWS 계정과 전달 규칙 공유 및 공유 규칙 사용

- 자세한 API 내용은 명령 참조PutResolverRulePolicy의 섹션을 참조하세요. AWS CLI

### `tag-resource`

다음 코드 예시에서는 `tag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

Resolver 리소스에 태그를 연결하려면

다음 `tag-resource` 예제에서는 두 태그 키/값 페어를 지정된 Resolver 규칙과 연결합니다.

```
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"
```

이 명령은 출력을 생성하지 않습니다.

비용 할당에 태그를 사용하는 방법에 대한 자세한 내용은 AWS 결제 및 비용 관리 사용 설명서의 비용 할당 태그 사용을 참조하세요.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### untag-resource

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

Resolver 리소스에서 태그를 제거하려면

다음 untag-resource 예제에서는 지정된 Resolver 규칙에서 두 개의 태그를 제거합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다. 태그가 제거되었는지 확인하려면 를 사용할 수 있습니다list-tags-for-resource.

비용 할당에 태그를 사용하는 방법에 대한 자세한 내용은 AWS 결제 및 비용 관리 사용 설명서의 비용 할당 태그 사용을 참조하세요.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-firewall-config

다음 코드 예시에서는 update-firewall-config을 사용하는 방법을 보여 줍니다.

AWS CLI

방화벽 구성을 업데이트하려면

다음 update-firewall-config 예제에서는 DNS 방화벽 구성을 업데이트합니다.

```
aws route53resolver update-firewall-config \
    --resource-id vpc-31e92222 \
    --firewall-fail-open DISABLED
```

출력:

```
{
    "FirewallConfig": {
        "Id": "rslvr-fc-86016850cexample",
        "ResourceId": "vpc-31e92222",
        "OwnerId": "123456789012",
        "FirewallFailOpen": "DISABLED"
    }
}
```

자세한 내용은 Amazon Route 53 개발자 안내서의 DNS 방화벽 VPC 구성을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateFirewallConfig의 섹션을 참조하세요. AWS CLI

### update-firewall-domains

다음 코드 예시에서는 update-firewall-domains을 사용하는 방법을 보여 줍니다.

AWS CLI

도메인 목록을 업데이트하려면

다음 update-firewall-domains 예제에서는 사용자가 제공한 ID로 도메인을 도메인 목록에 추가합니다.

```
aws route53resolver update-firewall-domains \
    --firewall-domain-list-id rslvr-fdl-42b60677cexampleb \
    --operation ADD \
    --domains test1.com test2.com test3.com
```

출력:

```
{
    "Id": "rslvr-fdl-42b60677cexample",
    "Name": "test",
```

```
    "Status": "UPDATING",
    "StatusMessage": "Updating the Firewall Domain List"
}
```

자세한 내용은 Amazon Route 53 개발자 안내서의 자체 도메인 목록 관리를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateFirewallDomains의 섹션을 참조하세요. AWS CLI

### update-firewall-rule-group-association

다음 코드 예시에서는 update-firewall-rule-group-association을 사용하는 방법을 보여 줍니다.

AWS CLI

방화벽 규칙 그룹 연결을 업데이트하려면

다음 update-firewall-rule-group-association 예제에서는 방화벽 규칙 그룹 연결을 업데이트합니다.

```
aws route53resolver update-firewall-rule-group-association \
    --firewall-rule-group-association-id rslvr-frgassoc-57e8873d7example \
    --priority 103
```

출력:

```
{
    "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"
    }
```

```
}
```

자세한 내용은 Amazon Route 53 개발자 안내서의 VPC 및 Route 53 Resolver DNS Firewall 규칙 그룹 간의 연결 관리를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateFirewallRuleGroupAssociation의 섹션을 참조하세요. AWS CLI

### update-firewall-rule

다음 코드 예시에서는 update-firewall-rule을 사용하는 방법을 보여 줍니다.

AWS CLI

방화벽 규칙을 업데이트하려면

다음 update-firewall-rule 예제에서는 방화벽 규칙을 지정한 파라미터로 업데이트합니다.

```
aws route53resolver update-firewall-rule \
    --firewall-rule-group-id rslvr-frg-47f93271fexample \
    --firewall-domain-list-id rslvr-fdl-9e956e9ffexample \
    --priority 102
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Route 53 개발자 안내서의 DNS 방화벽에서 규칙 그룹 및 규칙 관리를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateFirewallRule의 섹션을 참조하세요. AWS CLI

### update-resolver-endpoint

다음 코드 예시에서는 update-resolver-endpoint을 사용하는 방법을 보여 줍니다.

AWS CLI

Resolver 엔드포인트의 이름을 업데이트하려면

다음 update-resolver-endpoint 예제에서는 Resolver 엔드포인트의 이름을 업데이트합니다. 다른 값 업데이트는 지원되지 않습니다.

```
aws route53resolver update-resolver-endpoint \
    --resolver-endpoint-id rslvr-in-b5d45e32bdc445f09 \
    --name my-renamed-inbound-endpoint
```

출력:

```
{
    "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"
    }
}
```

- 자세한 API 내용은 명령 참조UpdateResolverEndpoint의 섹션을 참조하세요. AWS CLI

### update-resolver-rule

다음 코드 예시에서는 update-resolver-rule을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 설정 Resolver 엔드포인트 업데이트

다음 update-resolver-rule 예제에서는 규칙의 이름, DNS 쿼리가 전달되는 온프레미스 네트워크의 IP 주소, 쿼리를 네트워크에 전달하는 데 사용하는 아웃바운드 Resolver 엔드포인트의 ID를 업데이트합니다.

참고 TargetIps 의 기존 값을 덮어쓰므로 업데이트 후 규칙에 포함할 모든 IP 주소를 지정해야 합니다.

```
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
```

출력:

```
{
    "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"
    }
}
```

예제 2: ``config`` 설정에 대한 파일을 사용하여 설정 Resolver 엔드포인트를 업데이트하려면

또는 JSON 파일에 config 설정을 포함시킨 다음 를 호출할 때 해당 파일을 지정할 수 있습니다update-resolver-rule.

```
aws route53resolver update-resolver-rule \
    --resolver-rule-id rslvr-rr-1247fa64f3example \
    --config file://c:\temp\update-resolver-rule.json
```

update-resolver-rule.json의 콘텐츠.

```
{
    "Name": "my-2nd-rule",
    "TargetIps": [
        {
            "Ip": "192.0.2.45",
            "Port": 53
        },
        {
            "Ip": "192.0.2.46",
            "Port": 53
        }
    ],
    "ResolverEndpointId": "rslvr-out-7b89ed0d25example"
}
```

자세한 내용은 Amazon Route 53 개발자 안내서의 규칙을 생성하거나 편집할 때 지정하는 값을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateResolverRule의 섹션을 참조하세요. AWS CLI

## 를 사용한 Amazon S3 예제 AWS CLI

다음 코드 예제에서는 Amazon S3와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### abort-multipart-upload

다음 코드 예시에서는 abort-multipart-upload을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 멀티파트 업로드를 중단하려면

다음 abort-multipart-upload 명령은 my-bucket 버킷의 multipart/01 키에 대한 멀티파트 업로드를 중단합니다.

```
aws s3api abort-multipart-upload \
    --bucket my-bucket \
    --key multipart/01 \
    --upload-
id dfRtDYU0WWCCcH43C3WFbkRONycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3UR
```

이 명령에 필요한 업로드 ID는 create-multipart-upload에서 출력되며 list-multipart-uploads를 사용하여 검색할 수도 있습니다.

- 자세한 API 내용은 명령 참조AbortMultipartUpload의 섹션을 참조하세요. AWS CLI

### complete-multipart-upload

다음 코드 예시에서는 complete-multipart-upload을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 my-bucket 버킷의 multipart/01 키에 대한 멀티파트 업로드를 완료합니다.

```
aws s3api complete-multipart-upload --multipart-upload file://
mpustruct --bucket my-bucket --key 'multipart/01' --upload-
id dfRtDYU0WWCCcH43C3WFbkRONycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3UR
```

이 명령에 필요한 업로드 ID는 create-multipart-upload에서 출력되며 list-multipart-uploads를 사용하여 검색할 수도 있습니다.

위 명령의 멀티파트 업로드 옵션은 전체 파일로 재조립해야 하는 멀티파트 업로드의 부분을 설명하는 JSON 구조를 취합니다. 이 예제에서는 file:// 접두사를 사용하여 라는 로컬 폴더의 파일에서 JSON 구조를 로드합니다mpustruct.

mpustruct:

```
{
  "Parts": [
    {
      "ETag": "e868e0f4719e394144ef36531ee6824c",
      "PartNumber": 1
    },
    {
      "ETag": "6bb2b12753d66fe86da4998aa33fffb0",
      "PartNumber": 2
    },
    {
      "ETag": "d0a0112e841abec9c9ec83406f0159c8",
      "PartNumber": 3
    }
  ]
}
```

각 파트의 ETag 값은 upload-part 명령을 사용하여 파트를 업로드할 때마다 출력되며 를 호출하거나 각 파트의 MD5 체크섬을 가져와list-parts서 계산할 수도 있습니다.

출력:

```
{
    "ETag": "\"3944a9f7a4faab7f78788ff6210f63f0-3\"",
    "Bucket": "my-bucket",
    "Location": "https://my-bucket.s3.amazonaws.com/multipart%2F01",
    "Key": "multipart/01"
}
```

- 자세한 API 내용은 명령 참조CompleteMultipartUpload의 섹션을 참조하세요. AWS CLI

### copy-object

다음 코드 예시에서는 copy-object을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 bucket-1에서 bucket-2로 객체를 복사합니다.

```
aws s3api copy-object --copy-source bucket-1/test.txt --key test.txt --
bucket bucket-2
```

출력:

```
{
    "CopyObjectResult": {
        "LastModified": "2015-11-10T01:07:25.000Z",
        "ETag": "\"589c8b79c230a6ecd5a7e1d040a9a030\""
    },
    "VersionId": "YdnYvTCVDqRRFA.NFJjy36p0hxifMlkA"
}
```

- 자세한 API 내용은 명령 참조CopyObject의 섹션을 참조하세요. AWS CLI

### cp

다음 코드 예시에서는 cp을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 로컬 파일을 S3에 복사

다음 cp 명령은 단일 파일을 지정된 버킷 및 키에 복사합니다.

```
aws s3 cp test.txt s3://mybucket/test2.txt
```

출력:

```
upload: test.txt to s3://mybucket/test2.txt
```

예제 2: 만료 날짜가 있는 로컬 파일을 S3에 복사

다음 cp 명령은 지정된 ISO 8601 타임스탬프에 만료되는 단일 파일을 지정된 버킷 및 키에 복사합니다.

```
aws s3 cp test.txt s3://mybucket/test2.txt \
    --expires 2014-10-01T20:30:00Z
```

출력:

```
upload: test.txt to s3://mybucket/test2.txt
```

예제 3: S3에서 S3로 파일 복사 S3

다음 cp 명령은 단일 s3 객체를 지정된 버킷 및 키에 복사합니다.

```
aws s3 cp s3://mybucket/test.txt s3://mybucket/test2.txt
```

출력:

```
copy: s3://mybucket/test.txt to s3://mybucket/test2.txt
```

예제 4: 로컬 파일에 S3 객체 복사

다음 cp 명령은 단일 객체를 지정된 파일에 로컬로 복사합니다.

```
aws s3 cp s3://mybucket/test.txt test2.txt
```

출력:

```
download: s3://mybucket/test.txt to test2.txt
```

예제 5: 한 버킷에서 다른 버킷으로 S3 객체 복사

다음 cp 명령은 원래 이름을 유지하면서 단일 객체를 지정된 버킷에 복사합니다.

```
aws s3 cp s3://mybucket/test.txt s3://mybucket2/
```

출력:

```
copy: s3://mybucket/test.txt to s3://mybucket2/test.txt
```

예제 6: S3 객체를 로컬 디렉터리에 반복적으로 복사

파라미터 와 함께 전달되면 다음 cp 명령--recursive은 지정된 접두사 및 버킷에 있는 모든 객체를 지정된 디렉터리에 반복적으로 복사합니다. 이 예제에서는 버킷에 객체 test1.txt 및 mybucket가 있습니다test2.txt.

```
aws s3 cp s3://mybucket . \
    --recursive
```

출력:

```
download: s3://mybucket/test1.txt to test1.txt
download: s3://mybucket/test2.txt to test2.txt
```

예제 7: 로컬 파일을 S3에 반복적으로 복사

파라미터 와 함께 전달되면 --recursive다음 cp 명령은 파라미터를 사용하여 일부 파일을 제외하면서 지정된 디렉터리에 있는 모든 파일을 지정된 버킷 및 접두사에 반복적으로 복사합니다--exclude. 이 예제에서는 디렉터리에 파일 test1.txt 및 myDir가 있습니다test2.jpg.

```
aws s3 cp myDir s3://mybucket/ \
    --recursive \
    --exclude "*.jpg"
```

출력:

```
upload: myDir/test1.txt to s3://mybucket/test1.txt
```

예제 8: S3 객체를 다른 버킷에 반복적으로 복사

파라미터 와 함께 전달되면 다음 cp 명령은 --exclude 파라미터를 사용하여 일부 객체--recursive를 제외하면서 지정된 버킷에 있는 모든 객체를 다른 버킷에 반복적으로 복사합니다. 이 예제에서는 버킷에 객체 test1.txt 및 mybucket가 있습니다another/test1.txt.

```
aws s3 cp s3://mybucket/ s3://mybucket2/ \
    --recursive \
```

```
    --exclude "another/*"
```

출력:

```
copy: s3://mybucket/test1.txt to s3://mybucket2/test1.txt
```

--exclude 및 --include 옵션을 결합하여 다른 모든 객체를 제외하고 패턴과 일치하는 객체만 복사할 수 있습니다.

```
aws s3 cp s3://mybucket/logs/ s3://mybucket2/logs/ \
    --recursive \
    --exclude "*" \
    --include "*.log"
```

출력:

```
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
```

예제 9: S3 객체를 복사하는 동안 액세스 제어 목록(ACL) 설정

다음 cp 명령은 를 로 설정하는 동안 단일 객체를 지정된 버킷 및 키에 복사ACL합니다public-read-write.

```
aws s3 cp s3://mybucket/test.txt s3://mybucket/test2.txt \
    --acl public-read-write
```

출력:

```
copy: s3://mybucket/test.txt to s3://mybucket/test2.txt
```

--acl 옵션을 사용하는 경우 관련 IAM 정책에 "s3:PutObjectAcl" 작업이 포함되어 있는지 확인합니다.

```
aws iam get-user-policy \
    --user-name myuser \
    --policy-name mypolicy
```

출력:

```
{
    "UserName": "myuser",
    "PolicyName": "mypolicy",
    "PolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Action": [
                    "s3:PutObject",
                    "s3:PutObjectAcl"
                ],
                "Resource": [
                    "arn:aws:s3:::mybucket/*"
                ],
                "Effect": "Allow",
                "Sid": "Stmt1234567891234"
            }
        ]
    }
}
```

예제 10: S3 객체에 대한 권한 부여

다음 cp 명령은 에서 식별한 모든 사용자에게 읽기 액세스 권한을 부여URI하고 표준 ID로 식별한 특정 사용자에게 전체 제어 권한을 부여하는 --grants 옵션의 사용을 보여줍니다.

```
aws s3 cp file.txt s3://mybucket/ --grants read=uri=http://acs.amazonaws.com/groups/
global/
AllUsers full=id=79a59df900b949e55d96a1e698fbacedfd6e09d98eacf8f8d5218e7cd47ef2be
```

출력:

```
upload: file.txt to s3://mybucket/file.txt
```

예제 11: S3에 로컬 파일 스트림 업로드

PowerShell 는 의 인코딩을 변경하거나 파이프 입력에 를 추가할 CRLF 수 있습니다.

다음 cp 명령은 표준 입력에서 지정된 버킷 및 키로 로컬 파일 스트림을 업로드합니다.

```
aws s3 cp - s3://mybucket/stream.txt
```

예제 12: 50GB보다 큰 로컬 파일 스트림을 S3에 업로드

다음 cp 명령은 표준 입력에서 지정된 버킷 및 키로 51GB 로컬 파일 스트림을 업로드합니다. --expected-size 옵션을 제공해야 합니다. 그렇지 않으면 기본 부품 한도인 10,000에 도달하면 업로드가 실패할 수 있습니다.

```
aws s3 cp - s3://mybucket/stream.txt --expected-size 54760833024
```

예제 13: S3 객체를 로컬 파일 스트림으로 다운로드

PowerShell 는 의 인코딩을 변경하거나 파이프 또는 리디렉션된 출력CRLF에 를 추가할 수 있습니다.

다음 cp 명령은 S3 객체를 스트림으로 로컬에서 표준 출력으로 다운로드합니다. 스트림으로 다운로드는 현재 --recursive 파라미터와 호환되지 않습니다.

```
aws s3 cp s3://mybucket/stream.txt -
```

예제 14: S3 액세스 포인트에 업로드

다음 cp 명령은 키(mydoc.txt)의 액세스 포인트()에 단일 파일(myaccesspoint)을 업로드합니다mykey.

```
aws s3 cp mydoc.txt s3://arn:aws:s3:us-west-2:123456789012:accesspoint/
myaccesspoint/mykey
```

출력:

```
upload: mydoc.txt to s3://arn:aws:s3:us-west-2:123456789012:accesspoint/
myaccesspoint/mykey
```

예제 15: S3 액세스 포인트에서 다운로드

다음 cp 명령은 액세스 포인트(mykey)에서 로컬 파일()로 단일 객체(myaccesspoint)를 다운로드합니다mydoc.txt.

```
aws s3 cp s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/
mykey mydoc.txt
```

출력:

```
download: s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/mykey to
 mydoc.txt
```

- API 자세한 내용은 AWS CLI 명령 참조의 Cp를 참조하세요.

### `create-bucket`

다음 코드 예시에서는 `create-bucket`을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: 버킷을 생성하는 방법

다음 `create-bucket` 예시에서는 `my-bucket`이라는 버킷을 생성합니다.

```
aws s3api create-bucket \
    --bucket my-bucket \
    --region us-east-1
```

출력:

```
{
    "Location": "/my-bucket"
}
```

자세한 내용은 Amazon S3 사용 설명서의 버킷 생성을 참조하세요.

예 2: 소유자가 적용된 버킷을 생성하는 방법

다음 `create-bucket` 예시에서는 S3 객체 소유권에 대해 버킷 소유자 적용 설정을 사용하는 이름이 `my-bucket`인 버킷을 생성합니다.

```
aws s3api create-bucket \
    --bucket my-bucket \
    --region us-east-1 \
    --object-ownership BucketOwnerEnforced
```

출력:

```
{
    "Location": "/my-bucket"
}
```

자세한 내용은 Amazon S3 사용 설명서의 객체 소유권 제어 및 비활성화ACLs를 참조하세요.

예 3: ``us-east-1`` 리전 외부에서 버킷을 생성하는 방법

다음 create-bucket 예시에서는 eu-west-1 리전에서 my-bucket이라는 버킷을 생성합니다. us-east-1 외부 리전의 경우 원하는 리전에 버킷을 생성하려면 적절한 LocationConstraint를 지정해야 합니다.

```
aws s3api create-bucket \
    --bucket my-bucket \
    --region eu-west-1 \
    --create-bucket-configuration LocationConstraint=eu-west-1
```

출력:

```
{
    "Location": "http://my-bucket.s3.amazonaws.com/"
}
```

자세한 내용은 Amazon S3 사용 설명서의 버킷 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateBucket의 섹션을 참조하세요. AWS CLI

### create-multipart-upload

다음 코드 예시에서는 create-multipart-upload을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 multipart/01 키를 사용하여 my-bucket 버킷에 멀티파트 업로드를 생성합니다.

```
aws s3api create-multipart-upload --bucket my-bucket --key 'multipart/01'
```

출력:

```
{
```

```
    "Bucket": "my-bucket",
    "UploadId":
 "dfRtDYU0WWCCcH43C3WFbkRONycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3URC
    "Key": "multipart/01"
}
```

완성된 파일은 이름이 01이며 my-bucket 버킷의 multipart 폴더에 있습니다. upload-part 명령과 함께 사용할 업로드 ID, 키, 버킷 이름을 저장합니다.

- 자세한 API 내용은 명령 참조CreateMultipartUpload의 섹션을 참조하세요. AWS CLI

### delete-bucket-analytics-configuration

다음 코드 예시에서는 delete-bucket-analytics-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

버킷의 분석 구성을 삭제하려면

다음 delete-bucket-analytics-configuration 예시는 지정된 버킷 및 ID에 대한 분석 구성을 제거합니다.

```
aws s3api delete-bucket-analytics-configuration \
    --bucket my-bucket \
    --id 1
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeleteBucketAnalyticsConfiguration의 섹션을 참조하세요. AWS CLI

### delete-bucket-cors

다음 코드 예시에서는 delete-bucket-cors을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 이름이 my-bucket인 버킷에서 Cross-Origin Resource Sharing 구성을 삭제합니다.

```
aws s3api delete-bucket-cors --bucket my-bucket
```

- 자세한 API 내용은 명령 참조DeleteBucketCors의 섹션을 참조하세요. AWS CLI

### `delete-bucket-encryption`

다음 코드 예시에서는 `delete-bucket-encryption`을 사용하는 방법을 보여 줍니다.

AWS CLI

버킷의 서버 측 암호화 구성을 삭제하려면

다음 `delete-bucket-encryption` 예시는 지정된 버킷의 서버 측 암호화 구성을 삭제합니다.

```
aws s3api delete-bucket-encryption \
    --bucket my-bucket
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeleteBucketEncryption의 섹션을 참조하세요. AWS CLI

### `delete-bucket-intelligent-tiering-configuration`

다음 코드 예시에서는 `delete-bucket-intelligent-tiering-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

버킷에서 S3 Intelligent-Tiering 구성을 제거하려면

다음 `delete-bucket-intelligent-tiering-configuration` 예제에서는 버킷 ExampleConfig에서 라는 S3 Intelligent-Tiering 구성을 제거합니다.

```
aws s3api delete-bucket-intelligent-tiering-configuration \
    --bucket DOC-EXAMPLE-BUCKET \
    --id ExampleConfig
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon S3 사용 설명서의 S3 Intelligent-Tiering 사용을 참조하세요. Amazon S3

- 자세한 API 내용은 명령 참조DeleteBucketIntelligentTieringConfiguration의 섹션을 참조하세요. AWS CLI

### delete-bucket-inventory-configuration

다음 코드 예시에서는 delete-bucket-inventory-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

버킷의 인벤토리 구성을 삭제하려면

다음 delete-bucket-inventory-configuration 예시는 지정된 버킷에 대해 ID가 1인 인벤토리 구성을 삭제합니다.

```
aws s3api delete-bucket-inventory-configuration \
    --bucket my-bucket \
    --id 1
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeleteBucketInventoryConfiguration의 섹션을 참조하세요. AWS CLI

### delete-bucket-lifecycle

다음 코드 예시에서는 delete-bucket-lifecycle을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 my-bucket이라는 버킷에서 수명 주기 구성을 삭제합니다.

```
aws s3api delete-bucket-lifecycle --bucket my-bucket
```

- 자세한 API 내용은 명령 참조DeleteBucketLifecycle의 섹션을 참조하세요. AWS CLI

### delete-bucket-metrics-configuration

다음 코드 예시에서는 delete-bucket-metrics-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

버킷의 지표 구성을 삭제하려면

다음 delete-bucket-metrics-configuration 예시는 지정된 버킷 및 ID에 대한 지표 구성을 제거합니다.

```
aws s3api delete-bucket-metrics-configuration \
    --bucket my-bucket \
    --id 123
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeleteBucketMetricsConfiguration의 섹션을 참조하세요. AWS CLI

### delete-bucket-ownership-controls

다음 코드 예시에서는 delete-bucket-ownership-controls을 사용하는 방법을 보여 줍니다.

AWS CLI

버킷의 버킷 소유권 설정을 제거하려면

다음 delete-bucket-ownership-controls 예제에서는 버킷의 버킷 소유권 설정을 제거합니다.

```
aws s3api delete-bucket-ownership-controls \
    --bucket DOC-EXAMPLE-BUCKET
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon S3 사용 설명서의 기존 버킷에서 객체 소유권 설정을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteBucketOwnershipControls의 섹션을 참조하세요. AWS CLI

### delete-bucket-policy

다음 코드 예시에서는 delete-bucket-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 이름이 my-bucket인 버킷에서 버킷 정책을 삭제합니다.

```
aws s3api delete-bucket-policy --bucket my-bucket
```

- 자세한 API 내용은 명령 참조DeleteBucketPolicy의 섹션을 참조하세요. AWS CLI

### delete-bucket-replication

다음 코드 예시에서는 delete-bucket-replication을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 my-bucket이라는 버킷에서 복제 구성을 삭제합니다.

```
aws s3api delete-bucket-replication --bucket my-bucket
```

- 자세한 API 내용은 명령 참조DeleteBucketReplication의 섹션을 참조하세요. AWS CLI

### delete-bucket-tagging

다음 코드 예시에서는 delete-bucket-tagging을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 my-bucket이라는 버킷에서 태그 지정 구성을 삭제합니다.

```
aws s3api delete-bucket-tagging --bucket my-bucket
```

- 자세한 API 내용은 명령 참조DeleteBucketTagging의 섹션을 참조하세요. AWS CLI

### delete-bucket-website

다음 코드 예시에서는 delete-bucket-website을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 my-bucket이라는 버킷에서 웹 사이트 구성을 삭제합니다.

```
aws s3api delete-bucket-website --bucket my-bucket
```

- 자세한 API 내용은 명령 참조DeleteBucketWebsite의 섹션을 참조하세요. AWS CLI

### delete-bucket

다음 코드 예시에서는 delete-bucket을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 이름이 my-bucket인 버킷을 삭제합니다.

```
aws s3api delete-bucket --bucket my-bucket --region us-east-1
```

- 자세한 API 내용은 명령 참조DeleteBucket의 섹션을 참조하세요. AWS CLI

### delete-object-tagging

다음 코드 예시에서는 delete-object-tagging을 사용하는 방법을 보여 줍니다.

AWS CLI

객체의 태그 세트를 삭제하려면

다음 delete-object-tagging 예시는 지정된 키가 있는 태그를 doc1.rtf 객체에서 삭제합니다.

```
aws s3api delete-object-tagging \
    --bucket my-bucket \
    --key doc1.rtf
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeleteObjectTagging의 섹션을 참조하세요. AWS CLI

### delete-object

다음 코드 예시에서는 delete-object을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 이름이 my-bucket인 버킷에서 test.txt라는 객체를 삭제합니다.

```
aws s3api delete-object --bucket my-bucket --key test.txt
```

버킷 버전 관리가 활성화된 경우 출력에는 삭제 마커의 버전 ID가 포함됩니다.

```
{
  "VersionId": "9_gKg5vG56F.TTEUdwkxGpJ3tNDlWlGq",
```

```
  "DeleteMarker": true
}
```

객체 삭제에 대한 자세한 내용은 Amazon S3 개발자 안내서의 객체 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteObject의 섹션을 참조하세요. AWS CLI

### `delete-objects`

다음 코드 예시에서는 `delete-objects`을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 이름이 `my-bucket`인 버킷에서 객체를 삭제합니다.

```
aws s3api delete-objects --bucket my-bucket --delete file://delete.json
```

`delete.json` 는 삭제할 객체를 지정하는 현재 디렉터리의 JSON 문서입니다.

```
{
  "Objects": [
    {
      "Key": "test1.txt"
    }
  ],
  "Quiet": false
}
```

출력:

```
{
    "Deleted": [
        {
            "DeleteMarkerVersionId": "mYAT5Mc6F7aeUL8SS7FAAqUPO1koHwzU",
            "Key": "test1.txt",
            "DeleteMarker": true
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DeleteObjects의 섹션을 참조하세요. AWS CLI

### delete-public-access-block

다음 코드 예시에서는 delete-public-access-block을 사용하는 방법을 보여 줍니다.

AWS CLI

버킷의 퍼블릭 액세스 차단 구성을 삭제하려면

다음 delete-public-access-block 예시는 지정된 버킷에서 퍼블릭 액세스 차단 구성을 제거합니다.

```
aws s3api delete-public-access-block \
    --bucket my-bucket
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeletePublicAccessBlock의 섹션을 참조하세요. AWS CLI

### get-bucket-accelerate-configuration

다음 코드 예시에서는 get-bucket-accelerate-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

버킷의 가속화 구성을 검색하려면

다음 get-bucket-accelerate-configuration 예시는 지정된 버킷에 대한 가속 구성을 검색합니다.

```
aws s3api get-bucket-accelerate-configuration \
    --bucket my-bucket
```

출력:

```
{
    "Status": "Enabled"
}
```

- 자세한 API 내용은 명령 참조GetBucketAccelerateConfiguration의 섹션을 참조하세요. AWS CLI

### get-bucket-acl

다음 코드 예시에서는 get-bucket-acl을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 이름이 my-bucket인 버킷의 액세스 제어 목록을 검색합니다.

```
aws s3api get-bucket-acl --bucket my-bucket
```

출력:

```
{
    "Owner": {
        "DisplayName": "my-username",
        "ID": "7009a8971cd538e11f6b6606438875e7c86c5b672f46db45460ddcd087d36c32"
    },
    "Grants": [
        {
            "Grantee": {
                "DisplayName": "my-username",
                "ID":
 "7009a8971cd538e11f6b6606438875e7c86c5b672f46db45460ddcd087d36c32"
            },
            "Permission": "FULL_CONTROL"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetBucketAcl의 섹션을 참조하세요. AWS CLI

### get-bucket-analytics-configuration

다음 코드 예시에서는 get-bucket-analytics-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

특정 ID를 가진 버킷의 분석 구성을 검색하려면

다음 get-bucket-analytics-configuration 예시는 지정된 버킷 및 ID에 대한 분석 구성을 표시합니다.

```
aws s3api get-bucket-analytics-configuration \
    --bucket my-bucket \
    --id 1
```

출력:

```
{
    "AnalyticsConfiguration": {
        "StorageClassAnalysis": {},
        "Id": "1"
    }
}
```

- 자세한 API 내용은 명령 참조GetBucketAnalyticsConfiguration의 섹션을 참조하세요. AWS CLI

### get-bucket-cors

다음 코드 예시에서는 get-bucket-cors을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 이름이 my-bucket인 버킷에 대한 Cross-Origin Resource Sharing 구성을 검색합니다.

```
aws s3api get-bucket-cors --bucket my-bucket
```

출력:

```
{
    "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": [
                "*"
            ]
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetBucketCors의 섹션을 참조하세요. AWS CLI

### get-bucket-encryption

다음 코드 예시에서는 get-bucket-encryption을 사용하는 방법을 보여 줍니다.

AWS CLI

버킷의 서버 측 암호화 구성을 검색하려면

다음 get-bucket-encryption 예시는 my-bucket 버킷의 서버 측 암호화 구성을 검색합니다.

```
aws s3api get-bucket-encryption \
    --bucket my-bucket
```

출력:

```
{
    "ServerSideEncryptionConfiguration": {
        "Rules": [
            {
                "ApplyServerSideEncryptionByDefault": {
```

```
                    "SSEAlgorithm": "AES256"
                }
            }
        ]
    }
}
```

- 자세한 API 내용은 명령 참조GetBucketEncryption의 섹션을 참조하세요. AWS CLI

### `get-bucket-intelligent-tiering-configuration`

다음 코드 예시에서는 `get-bucket-intelligent-tiering-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

버킷에서 S3 Intelligent-Tiering 구성을 검색하려면

다음 `get-bucket-intelligent-tiering-configuration` 예제는 버킷 ExampleConfig에서라는 S3 Intelligent-Tiering 구성을 검색합니다.

```
aws s3api get-bucket-intelligent-tiering-configuration \
    --bucket DOC-EXAMPLE-BUCKET \
    --id ExampleConfig
```

출력:

```
{
    "IntelligentTieringConfiguration": {
        "Id": "ExampleConfig2",
        "Filter": {
            "Prefix": "images"
        },
        "Status": "Enabled",
        "Tierings": [
            {
                "Days": 90,
                "AccessTier": "ARCHIVE_ACCESS"
            },
            {
                "Days": 180,
                "AccessTier": "DEEP_ARCHIVE_ACCESS"
```

```
            }
        ]
    }
}
```

자세한 내용은 Amazon S3 사용 설명서의 S3 Intelligent-Tiering 사용을 참조하세요.

- 자세한 API 내용은 AWS CLI 명령 참조의 GetBucketIntelligentTieringConfiguration 섹션을 참조하세요.

### get-bucket-inventory-configuration

다음 코드 예시에서는 get-bucket-inventory-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

버킷의 인벤토리 구성을 검색하려면

다음 get-bucket-inventory-configuration 예시는 지정된 버킷에 대해 ID가 1인 인벤토리 구성을 검색합니다.

```
aws s3api get-bucket-inventory-configuration \
    --bucket my-bucket \
    --id 1
```

출력:

```
{
    "InventoryConfiguration": {
        "IsEnabled": true,
        "Destination": {
            "S3BucketDestination": {
                "Format": "ORC",
                "Bucket": "arn:aws:s3:::my-bucket",
                "AccountId": "123456789012"
            }
        },
        "IncludedObjectVersions": "Current",
        "Id": "1",
        "Schedule": {
            "Frequency": "Weekly"
        }
    }
```

```
}
```

- 자세한 API 내용은 명령 참조GetBucketInventoryConfiguration의 섹션을 참조하세요. AWS CLI

### get-bucket-lifecycle-configuration

다음 코드 예시에서는 get-bucket-lifecycle-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 이름이 my-bucket인 버킷의 수명 주기 구성을 검색합니다.

```
aws s3api get-bucket-lifecycle-configuration --bucket my-bucket
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetBucketLifecycleConfiguration의 섹션을 참조하세요. AWS CLI

### get-bucket-lifecycle

다음 코드 예시에서는 get-bucket-lifecycle을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 이름이 my-bucket인 버킷의 수명 주기 구성을 검색합니다.

```
aws s3api get-bucket-lifecycle --bucket my-bucket
```

출력:

```
{
  "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"
    }
  ]
}
```

- 자세한 API 내용은 명령 참조GetBucketLifecycle의 섹션을 참조하세요. AWS CLI

### get-bucket-location

다음 코드 예시에서는 get-bucket-location을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 이름이 my-bucket인 버킷의 위치 제약 조건을 검색합니다(제약 조건이 있는 경우).

```
aws s3api get-bucket-location --bucket my-bucket
```

출력:

```
{
    "LocationConstraint": "us-west-2"
}
```

- 자세한 API 내용은 명령 참조GetBucketLocation의 섹션을 참조하세요. AWS CLI

### get-bucket-logging

다음 코드 예시에서는 get-bucket-logging을 사용하는 방법을 보여 줍니다.

AWS CLI

버킷의 로깅 상태를 검색하려면

다음 get-bucket-logging 예시는 지정된 버킷의 로깅 상태를 검색합니다.

```
aws s3api get-bucket-logging \
    --bucket my-bucket
```

출력:

```
{
    "LoggingEnabled": {
        "TargetPrefix": "",
        "TargetBucket": "my-bucket-logs"
          }
}
```

- 자세한 API 내용은 명령 참조GetBucketLogging의 섹션을 참조하세요. AWS CLI

### get-bucket-metrics-configuration

다음 코드 예시에서는 get-bucket-metrics-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

특정 ID를 가진 버킷의 지표 구성을 검색하려면

다음 `get-bucket-metrics-configuration` 예시는 지정된 버킷 및 ID에 대한 지표 구성을 표시합니다.

```
aws s3api get-bucket-metrics-configuration \
    --bucket my-bucket \
    --id 123
```

출력:

```
{
    "MetricsConfiguration": {
        "Filter": {
            "Prefix": "logs"
        },
        "Id": "123"
    }
}
```

- 자세한 API 내용은 명령 참조GetBucketMetricsConfiguration의 섹션을 참조하세요. AWS CLI

### `get-bucket-notification-configuration`

다음 코드 예시에서는 `get-bucket-notification-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 이름이 `my-bucket`인 버킷의 알림 구성을 검색합니다.

```
aws s3api get-bucket-notification-configuration --bucket my-bucket
```

출력:

```
{
    "TopicConfigurations": [
        {
            "Id": "YmQzMmEwM2EjZWVlI0NGItNzVtZjI1MC00ZjgyLWZDBiZWNl",
```

```
            "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-notification-topic",
            "Events": [
                "s3:ObjectCreated:*"
            ]
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetBucketNotificationConfiguration의 섹션을 참조하세요. AWS CLI

### get-bucket-notification

다음 코드 예시에서는 get-bucket-notification을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 이름이 my-bucket인 버킷의 알림 구성을 검색합니다.

```
aws s3api get-bucket-notification --bucket my-bucket
```

출력:

```
{
    "TopicConfiguration": {
        "Topic": "arn:aws:sns:us-west-2:123456789012:my-notification-topic",
        "Id": "YmQzMmEwM2EjZWVlI0NGItNzVtZjI1MC00ZjgyLWZDBiZWNl",
        "Event": "s3:ObjectCreated:*",
        "Events": [
            "s3:ObjectCreated:*"
        ]
    }
}
```

- 자세한 API 내용은 명령 참조GetBucketNotification의 섹션을 참조하세요. AWS CLI

### get-bucket-ownership-controls

다음 코드 예시에서는 get-bucket-ownership-controls을 사용하는 방법을 보여 줍니다.

AWS CLI

버킷의 버킷 소유권 설정을 검색하려면

다음 get-bucket-ownership-controls 예제에서는 버킷의 버킷 소유권 설정을 검색합니다.

```
aws s3api get-bucket-ownership-controls \
    --bucket DOC-EXAMPLE-BUCKET
```

출력:

```
{
    "OwnershipControls": {
        "Rules": [
            {
                "ObjectOwnership": "BucketOwnerEnforced"
            }
        ]
    }
}
```

자세한 내용은 Amazon S3 사용 설명서의 S3 버킷에 대한 객체 소유권 설정 보기를 참조하세요. Amazon S3

- 자세한 API 내용은 명령 참조GetBucketOwnershipControls의 섹션을 참조하세요. AWS CLI

### get-bucket-policy-status

다음 코드 예시에서는 get-bucket-policy-status을 사용하는 방법을 보여 줍니다.

AWS CLI

특정 버킷이 퍼블릭인지 나타내는 버킷 정책 상태를 검색하려면

다음 get-bucket-policy-status 예시는 my-bucket 버킷의 정책 상태를 검색합니다.

```
aws s3api get-bucket-policy-status \
    --bucket my-bucket
```

출력:

```
{
    "PolicyStatus": {
        "IsPublic": false
    }
```

```
}
```

- 자세한 API 내용은 명령 참조GetBucketPolicyStatus의 섹션을 참조하세요. AWS CLI

### get-bucket-policy

다음 코드 예시에서는 get-bucket-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 이름이 my-bucket인 버킷의 버킷 정책을 검색합니다.

```
aws s3api get-bucket-policy --bucket my-bucket
```

출력:

```
{
    "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/*\"}]}"
}
```

policyThe 다음 예제에서는 Amazon S3 버킷 정책을 다운로드하고 파일을 수정한 다음 put-bucket-policy를 사용하여 수정된 버킷 정책을 적용하는 방법을 보여 줍니다. 버킷 정책을 파일로 다운로드하려면 다음을 실행할 수 있습니다.

```
aws s3api get-bucket-policy --bucket mybucket --query Policy --output text >
 policy.json
```

그런 다음 필요에 따라 policy.json 파일을 수정할 수 있습니다. 마지막으로 필요에 따라

policy.json 파일을 실행하여 수정된 정책을 S3 버킷에 다시 적용할 수 있습니다. 마지막으로 필요에 따라

파일을 실행하여 수정된 정책을 S3 버킷에 다시 적용할 수 있습니다. 마지막으로 필요에 따라 다음을 실행하여 수정된 정책을 S3 버킷에 다시 적용할 수 있습니다.

```
aws s3api put-bucket-policy --bucket mybucket --policy file://policy.json
```

- 자세한 API 내용은 명령 참조GetBucketPolicy의 섹션을 참조하세요. AWS CLI

### get-bucket-replication

다음 코드 예시에서는 get-bucket-replication을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 이름이 my-bucket인 버킷의 복제 구성을 검색합니다.

```
aws s3api get-bucket-replication --bucket my-bucket
```

출력:

```
{
    "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"
    }
}
```

- 자세한 API 내용은 명령 참조GetBucketReplication의 섹션을 참조하세요. AWS CLI

### get-bucket-request-payment

다음 코드 예시에서는 get-bucket-request-payment을 사용하는 방법을 보여 줍니다.

AWS CLI

버킷의 지불 요청 구성을 검색하려면

다음 get-bucket-request-payment 예시는 지정된 버킷에 대한 요청자 지불 구성을 검색합니다.

```
aws s3api get-bucket-request-payment \
    --bucket my-bucket
```

출력:

```
{
    "Payer": "BucketOwner"
}
```

- 자세한 API 내용은 명령 참조GetBucketRequestPayment의 섹션을 참조하세요. AWS CLI

### get-bucket-tagging

다음 코드 예시에서는 get-bucket-tagging을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 이름이 my-bucket인 버킷에 대한 태그 지정 구성을 검색합니다.

```
aws s3api get-bucket-tagging --bucket my-bucket
```

출력:

```
{
    "TagSet": [
        {
            "Value": "marketing",
            "Key": "organization"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetBucketTagging의 섹션을 참조하세요. AWS CLI

### get-bucket-versioning

다음 코드 예시에서는 get-bucket-versioning을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 이름이 my-bucket인 버킷에 대한 버전 관리 구성을 검색합니다.

```
aws s3api get-bucket-versioning --bucket my-bucket
```

출력:

```
{
    "Status": "Enabled"
}
```

- 자세한 API 내용은 명령 참조GetBucketVersioning의 섹션을 참조하세요. AWS CLI

### get-bucket-website

다음 코드 예시에서는 get-bucket-website을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 이름이 my-bucket인 버킷의 정적 웹 사이트 구성을 검색합니다.

```
aws s3api get-bucket-website --bucket my-bucket
```

출력:

```
{
    "IndexDocument": {
        "Suffix": "index.html"
    },
    "ErrorDocument": {
        "Key": "error.html"
    }
}
```

- 자세한 API 내용은 명령 참조GetBucketWebsite의 섹션을 참조하세요. AWS CLI

### get-object-acl

다음 코드 예시에서는 get-object-acl을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 이름이 my-bucket인 버킷의 객체에 대한 액세스 제어 목록을 검색합니다.

```
aws s3api get-object-acl --bucket my-bucket --key index.html
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetObjectAcl의 섹션을 참조하세요. AWS CLI

### get-object-attributes

다음 코드 예시에서는 get-object-attributes을 사용하는 방법을 보여 줍니다.

AWS CLI

객체 자체를 반환하지 않고 객체에서 메타데이터를 검색하려면

다음 get-object-attributes 예시에서는 doc1.rtf 객체에서 메타데이터를 검색합니다.

```
aws s3api get-object-attributes \
    --bucket my-bucket \
    --key doc1.rtf \
    --object-attributes "StorageClass" "ETag" "ObjectSize"
```

출력:

```
{
    "LastModified": "2022-03-15T19:37:31+00:00",
    "VersionId": "IuCPjXTDzHNfldAuitVBIKJpF2p1fg4P",
    "ETag": "b662d79adeb7c8d787ea7eafb9ef6207",
    "StorageClass": "STANDARD",
    "ObjectSize": 405
}
```

자세한 내용은 Amazon S3 API 참조GetObjectAttributes의 섹션을 참조하세요.

- 자세한 API 내용은 명령 참조GetObjectAttributes의 섹션을 참조하세요. AWS CLI

### get-object-legal-hold

다음 코드 예시에서는 get-object-legal-hold을 사용하는 방법을 보여 줍니다.

AWS CLI

객체의 법적 보류 상태 검색

다음 get-object-legal-hold 예시는 지정된 객체의 법적 보류 상태를 검색합니다.

```
aws s3api get-object-legal-hold \
    --bucket my-bucket-with-object-lock \
    --key doc1.rtf
```

출력:

```
{
    "LegalHold": {
        "Status": "ON"
    }
}
```

- 자세한 API 내용은 명령 참조GetObjectLegalHold의 섹션을 참조하세요. AWS CLI

### get-object-lock-configuration

다음 코드 예시에서는 get-object-lock-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

버킷의 객체 잠금 구성을 검색하는 방법

다음 get-object-lock-configuration 예시에서는 지정된 버킷에 대한 객체 잠금 구성을 검색합니다.

```
aws s3api get-object-lock-configuration \
    --bucket my-bucket-with-object-lock
```

출력:

```
{
    "ObjectLockConfiguration": {
        "ObjectLockEnabled": "Enabled",
        "Rule": {
            "DefaultRetention": {
                "Mode": "COMPLIANCE",
                "Days": 50
            }
        }
    }
}
```

- 자세한 API 내용은 명령 참조GetObjectLockConfiguration의 섹션을 참조하세요. AWS CLI

### get-object-retention

다음 코드 예시에서는 get-object-retention을 사용하는 방법을 보여 줍니다.

AWS CLI

객체의 객체 보존 구성을 검색하는 방법

다음 get-object-retention 예시에서는 지정된 객체에 대한 보존 구성을 검색합니다.

```
aws s3api get-object-retention \
```

```
    --bucket my-bucket-with-object-lock \
    --key doc1.rtf
```

출력:

```
{
    "Retention": {
        "Mode": "GOVERNANCE",
        "RetainUntilDate": "2025-01-01T00:00:00.000Z"
    }
}
```

- 자세한 API 내용은 명령 참조GetObjectRetention의 섹션을 참조하세요. AWS CLI

### get-object-tagging

다음 코드 예시에서는 get-object-tagging을 사용하는 방법을 보여 줍니다.

AWS CLI

객체에 연결된 태그를 검색하려면

다음 get-object-tagging 예시는 지정된 객체에서 지정된 키의 값을 검색합니다.

```
aws s3api get-object-tagging \
    --bucket my-bucket \
    --key doc1.rtf
```

출력:

```
{
    "TagSet": [
        {
            "Value": "confidential",
            "Key": "designation"
        }
    ]
}
```

다음 get-object-tagging 예시는 태그가 없는 doc2.rtf 객체의 태그 세트를 검색하려고 시도합니다.

```
aws s3api get-object-tagging \
    --bucket my-bucket \
    --key doc2.rtf
```

출력:

```
{
    "TagSet": []
}
```

다음 get-object-tagging 예시는 태그가 여러 개 있는 doc3.rtf 객체의 태그 세트를 검색합니다.

```
aws s3api get-object-tagging \
    --bucket my-bucket \
    --key doc3.rtf
```

출력:

```
{
    "TagSet": [
        {
            "Value": "confidential",
            "Key": "designation"
        },
        {
            "Value": "finance",
            "Key": "department"
        },
        {
            "Value": "payroll",
            "Key": "team"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetObjectTagging의 섹션을 참조하세요. AWS CLI

### get-object-torrent

다음 코드 예시에서는 get-object-torrent을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 라는 버킷에 객체에 대한 토런트를 생성합니다my-bucket.

```
aws s3api get-object-torrent --bucket my-bucket --key large-video-file.mp4 large-
video-file.torrent
```

토렌트 파일은 현재 폴더에 로컬로 저장됩니다. 출력 파일 이름(large-video-file.torrent)은 옵션 이름 없이 지정되며 명령의 마지막 인수여야 합니다.

- 자세한 API 내용은 명령 참조GetObjectTorrent의 섹션을 참조하세요. AWS CLI

### get-object

다음 코드 예시에서는 get-object을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 예시에서는 get-object 명령을 사용하여 Amazon S3에서 객체를 다운로드합니다.

```
aws s3api get-object --bucket text-content --key dir/
my_images.tar.bz2 my_images.tar.bz2
```

참고로 outfile 파라미터는 “--outfile”과 같은 옵션 이름 없이 지정됩니다. 출력 파일의 이름은 명령의 마지막 파라미터여야 합니다.

아래 예시에서는 --range를 사용하여 객체에서 특정 바이트 범위를 다운로드하는 방법을 보여줍니다. 참고로 바이트 범위에는 “bytes=”라는 접두사가 있어야 합니다.

```
aws s3api get-object --bucket text-content --key dir/my_data --
range bytes=8888-9999 my_data_range
```

객체 검색에 대한 자세한 내용은 Amazon S3 개발자 안내서의 객체 가져오기를 참조하세요.

- 자세한 API 내용은 명령 참조GetObject의 섹션을 참조하세요. AWS CLI

### get-public-access-block

다음 코드 예시에서는 get-public-access-block을 사용하는 방법을 보여 줍니다.

AWS CLI

버킷의 퍼블릭 액세스 차단 구성을 설정하거나 수정하려면

다음 `get-public-access-block` 예시는 지정된 버킷에 대한 퍼블릭 액세스 차단 구성을 표시합니다.

```
aws s3api get-public-access-block \
    --bucket my-bucket
```

출력:

```
{
    "PublicAccessBlockConfiguration": {
        "IgnorePublicAcls": true,
        "BlockPublicPolicy": true,
        "BlockPublicAcls": true,
        "RestrictPublicBuckets": true
    }
}
```

- 자세한 API 내용은 명령 참조GetPublicAccessBlock의 섹션을 참조하세요. AWS CLI

### `head-bucket`

다음 코드 예시에서는 `head-bucket`을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 `my-bucket`이라는 버킷에 대한 액세스를 확인합니다.

```
aws s3api head-bucket --bucket my-bucket
```

버킷이 존재하고 버킷에 대한 액세스 권한이 있는 경우 출력이 반환되지 않습니다. 그렇지 않으면 오류 메시지가 표시됩니다. 예:

```
A client error (404) occurred when calling the HeadBucket operation: Not Found
```

- 자세한 API 내용은 명령 참조HeadBucket의 섹션을 참조하세요. AWS CLI

### head-object

다음 코드 예시에서는 head-object을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 my-bucket이라는 버킷의 객체에 대한 메타데이터를 검색합니다.

```
aws s3api head-object --bucket my-bucket --key index.html
```

출력:

```
{
    "AcceptRanges": "bytes",
    "ContentType": "text/html",
    "LastModified": "Thu, 16 Apr 2015 18:19:14 GMT",
    "ContentLength": 77,
    "VersionId": "null",
    "ETag": "\"30a6ec7e1a9ad79c203d05a589c8b400\"",
    "Metadata": {}
}
```

- 자세한 API 내용은 명령 참조HeadObject의 섹션을 참조하세요. AWS CLI

### list-bucket-analytics-configurations

다음 코드 예시에서는 list-bucket-analytics-configurations을 사용하는 방법을 보여 줍니다.

AWS CLI

버킷의 분석 구성 목록을 검색하려면

다음 list-bucket-analytics-configurations는 지정된 버킷에 대한 분석 구성 목록을 검색합니다.

```
aws s3api list-bucket-analytics-configurations \
    --bucket my-bucket
```

출력:

```
{
    "AnalyticsConfigurationList": [
        {
            "StorageClassAnalysis": {},
            "Id": "1"
        }
    ],
    "IsTruncated": false
}
```

- 자세한 API 내용은 명령 참조ListBucketAnalyticsConfigurations의 섹션을 참조하세요. AWS CLI

### `list-bucket-intelligent-tiering-configurations`

다음 코드 예시에서는 `list-bucket-intelligent-tiering-configurations`을 사용하는 방법을 보여 줍니다.

AWS CLI

버킷에서 모든 S3 Intelligent-Tiering 구성을 검색하려면

다음 `list-bucket-intelligent-tiering-configurations` 예제에서는 버킷의 모든 S3 Intelligent-Tiering 구성을 검색합니다.

```
aws s3api list-bucket-intelligent-tiering-configurations \
    --bucket DOC-EXAMPLE-BUCKET
```

출력:

```
{
    "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"
                }
            ]
        }
    ]
}
```

자세한 내용은 Amazon S3 사용 설명서의 S3 Intelligent-Tiering 사용을 참조하세요. Amazon S3

- 자세한 API 내용은 명령 참조ListBucketIntelligentTieringConfigurations의 섹션을 참조하세요. AWS CLI

### list-bucket-inventory-configurations

다음 코드 예시에서는 list-bucket-inventory-configurations을 사용하는 방법을 보여 줍니다.

AWS CLI

버킷의 인벤토리 구성을 검색하려면

다음 list-bucket-inventory-configurations 예시는 지정된 버킷의 인벤토리 구성을 나열합니다.

```
aws s3api list-bucket-inventory-configurations \
    --bucket my-bucket
```

출력:

```
{
    "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
}
```

- 자세한 API 내용은 명령 참조ListBucketInventoryConfigurations의 섹션을 참조하세요. AWS CLI

### `list-bucket-metrics-configurations`

다음 코드 예시에서는 `list-bucket-metrics-configurations`을 사용하는 방법을 보여 줍니다.

AWS CLI

버킷의 지표 구성 목록을 검색하려면

다음 `list-bucket-metrics-configurations` 예제에서는 지정된 버킷에 대한 지표 구성 목록을 검색합니다.

```
aws s3api list-bucket-metrics-configurations \
    --bucket my-bucket
```

출력:

```
{
    "IsTruncated": false,
    "MetricsConfigurationList": [
        {
            "Filter": {
                "Prefix": "logs"
            },
            "Id": "123"
        },
        {
            "Filter": {
                "Prefix": "tmp"
            },
            "Id": "234"
```

```
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListBucketMetricsConfigurations의 섹션을 참조하세요. AWS CLI

### `list-buckets`

다음 코드 예시에서는 `list-buckets`을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 `list-buckets` 명령을 사용하여 모든 Amazon S3 버킷(모든 리전)의 이름을 표시합니다.

```
aws s3api list-buckets --query "Buckets[].Name"
```

쿼리 옵션은 `list-buckets`의 출력을 버킷 이름으로만 필터링합니다.

버킷에 대한 자세한 내용은 Amazon S3 개발자 안내서의 Amazon S3 버킷 작업을 참조하세요.

- 자세한 API 내용은 명령 참조ListBuckets의 섹션을 참조하세요. AWS CLI

### `list-multipart-uploads`

다음 코드 예시에서는 `list-multipart-uploads`을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 이름이 `my-bucket`인 버킷의 활성 멀티파트 업로드를 모두 나열합니다.

```
aws s3api list-multipart-uploads --bucket my-bucket
```

출력:

```
{
    "Uploads": [
        {
            "Initiator": {
                "DisplayName": "username",
                "ID": "arn:aws:iam::0123456789012:user/username"
            },
```

```
            "Initiated": "2015-06-02T18:01:30.000Z",
            "UploadId":
 "dfRtDYU0WWCCcH43C3WFbkRONycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3URC
            "StorageClass": "STANDARD",
            "Key": "multipart/01",
            "Owner": {
                "DisplayName": "aws-account-name",
                "ID":
 "100719349fc3b6dcd7c820a124bf7aecd408092c3d7b51b38494939801fc248b"
            }
        }
    ],
    "CommonPrefixes": []
}
```

진행 중인 멀티파트 업로드는 Amazon S3에서 스토리지 비용을 발생시킵니다. 활성 멀티파트 업로드를 완료하거나 중단하여 계정에서 해당 파트를 제거하세요.

- 자세한 API 내용은 명령 참조ListMultipartUploads의 섹션을 참조하세요. AWS CLI

### `list-object-versions`

다음 코드 예시에서는 `list-object-versions`을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 이름이 `my-bucket`인 버킷의 객체의 버전 정보를 검색합니다.

```
aws s3api list-object-versions --bucket my-bucket --prefix index.html
```

출력:

```
{
    "DeleteMarkers": [
        {
            "Owner": {
                "DisplayName": "my-username",
                "ID":
 "7009a8971cd660687538875e7c86c5b672fe116bd438f46db45460ddcd036c32"
            },
            "IsLatest": true,
            "VersionId": "B2VsEK5saUNNHKcOAJj7hIE86RozToyq",
            "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListObjectVersions의 섹션을 참조하세요. AWS CLI

### `list-objects-v2`

다음 코드 예시에서는 `list-objects-v2`을 사용하는 방법을 보여 줍니다.

AWS CLI

버킷의 객체 목록을 보려면

다음 `list-objects-v2` 예시는 지정된 버킷의 객체를 나열합니다.

```
aws s3api list-objects-v2 \
    --bucket my-bucket
```

출력:

```
{
    "Contents": [
        {
            "LastModified": "2019-11-05T23:11:50.000Z",
            "ETag": "\"621503c373607d548b37cff8778d992c\"",
            "StorageClass": "STANDARD",
            "Key": "doc1.rtf",
            "Size": 391
        },
```

```
        {
            "LastModified": "2019-11-05T23:11:50.000Z",
            "ETag": "\"a2cecc36ab7c7fe3a71a273b9d45b1b5\"",
            "StorageClass": "STANDARD",
            "Key": "doc2.rtf",
            "Size": 373
        },
        {
            "LastModified": "2019-11-05T23:11:50.000Z",
            "ETag": "\"08210852f65a2e9cb999972539a64d68\"",
            "StorageClass": "STANDARD",
            "Key": "doc3.rtf",
            "Size": 399
        },
        {
            "LastModified": "2019-11-05T23:11:50.000Z",
            "ETag": "\"d1852dd683f404306569471af106988e\"",
            "StorageClass": "STANDARD",
            "Key": "doc4.rtf",
            "Size": 6225
        }
    ]
}
```

- API 자세한 내용은 AWS CLI 명령 참조의 ListObjectsV2를 참조하세요.

### `list-objects`

다음 코드 예시에서는 `list-objects`을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 예시에서는 `list-objects` 명령을 사용하여 지정된 버킷에 있는 모든 객체의 이름을 표시합니다.

```
aws s3api list-objects --bucket text-content --query 'Contents[].{Key: Key, Size:
 Size}'
```

이 예시에서는 `--query` 인수를 사용하여 `list-objects`의 출력을 각 객체의 키 값 및 크기로 필터링합니다.

객체에 대한 자세한 내용은 Amazon S3 개발자 안내서의 Amazon S3 객체 작업을 참조하세요.

- 자세한 API 내용은 명령 참조ListObjects의 섹션을 참조하세요. AWS CLI

### list-parts

다음 코드 예시에서는 list-parts을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 버킷 에 키가 있는 멀티파트 업로드에 대해 업로드된 모든 부분을 나열multipart/01합니다my-bucket.

```
aws s3api list-parts --bucket my-bucket --key 'multipart/01' --upload-
id dfRtDYU0WWCCcH43C3WFbkRONycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3UR
```

출력:

```
{
    "Owner": {
        "DisplayName": "aws-account-name",
        "ID": "100719349fc3b6dcd7c820a124bf7aecd408092c3d7b51b38494939801fc248b"
    },
    "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"
}
```

- 자세한 API 내용은 명령 참조ListParts의 섹션을 참조하세요. AWS CLI

### `ls`

다음 코드 예시에서는 `ls`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 사용자 소유 버킷 모두 나열

다음 `ls` 명령은 사용자가 소유한 모든 버킷을 나열합니다. 이 예제에서는 사용자가 버킷 `mybucket` 및 를 소유합니다`mybucket2`. 타임스탬프는 버킷이 생성된 날짜로, 제품의 시간대에 표시됩니다. 버킷 정책 편집과 같이 버킷을 변경할 때 이 날짜가 변경될 수 있습니다. `s3://` 가 경로 인수 에 사용되는 경우 모든 버킷도 나열`<S3Uri>`됩니다.

```
aws s3 ls
```

출력:

```
2013-07-11 17:08:50 mybucket
2013-07-24 14:55:44 mybucket2
```

예제 2: 버킷의 모든 접두사 및 객체 나열

다음 `ls` 명령은 지정된 버킷 및 접두사 아래에 객체와 공통 접두사를 나열합니다. 이 예제에서는 사용자가 객체 `test.txt` 및 를 `mybucket` 사용하여 버킷을 소유합니다`somePrefix/test.txt`. `LastWriteTime` 및 `Length`는 임의입니다. `ls` 명령은 로컬 파일 시스템과 상호 작용하지 않으므로 모호성을 해결하는 데 `s3://` URI 체계가 필요하지 않으며 생략될 수 있습니다.

```
aws s3 ls s3://mybucket
```

출력:

```
                           PRE somePrefix/
```

```
2013-07-25 17:06:27         88 test.txt
```

예제 3: 특정 버킷 및 접두사에 있는 모든 접두사 및 객체 나열

다음 `ls` 명령은 지정된 버킷 및 접두사 아래에 객체와 공통 접두사를 나열합니다. 그러나 지정된 버킷 및 접두사 아래에는 객체나 공통 접두사가 없습니다.

```
aws s3 ls s3://mybucket/noExistPrefix
```

출력:

```
None
```

예제 4: 버킷의 모든 접두사 및 객체를 반복적으로 나열

다음 `ls` 명령은 버킷의 객체를 반복적으로 나열합니다. 출력`PRE dirname/`에 표시하는 대신 버킷의 모든 콘텐츠가 순서대로 나열됩니다.

```
aws s3 ls s3://mybucket \
    --recursive
```

출력:

```
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
```

예제 5: 버킷의 모든 접두사 및 객체 요약

다음 `ls` 명령은 --인간이 읽을 수 있는 옵션과 --요약 옵션을 사용하여 동일한 명령을 보여줍니다. --인간이 읽을 수 있는 는 에 파일 크기를 표시합니다Bytes/MiB/KiB/GiB/TiB/PiB/EiB. --요약은 결과 목록 끝에 총 객체 수와 총 크기를 표시합니다.

```
aws s3 ls s3://mybucket \
    --recursive \
    --human-readable \
    --summarize
```

출력:

```
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
```

예제 6: S3 액세스 포인트에서 나열

다음 `ls` 명령은 액세스 포인트(`myaccesspoint`)의 객체를 나열합니다.

```
aws s3 ls s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/
```

출력:

```
                           PRE somePrefix/
2013-07-25 17:06:27         88 test.txt
```

- API 자세한 내용은 AWS CLI 명령 참조의 Ls를 참조하세요.

### mb

다음 코드 예시에서는 mb을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 버킷 생성

다음 mb 명령은 버킷을 생성합니다. 이 예제에서는 사용자가 버킷을 만듭니다mybucket. 버킷은 사용자의 구성 파일에 지정된 리전에 생성됩니다.

```
aws s3 mb s3://mybucket
```

출력:

```
make_bucket: s3://mybucket
```

예제 2: 지정된 리전에서 버킷 생성

다음 mb 명령은 --region 파라미터에 의해 지정된 리전에 버킷을 생성합니다. 이 예제에서는 사용자가 리전 mybucket 에서 버킷을 만듭니다us-west-1.

```
aws s3 mb s3://mybucket \
    --region us-west-1
```

출력:

```
make_bucket: s3://mybucket
```

- API 자세한 내용은 AWS CLI 명령 참조의 Mb를 참조하세요.

**mv**

다음 코드 예시에서는 mv을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 로컬 파일을 지정된 버킷으로 이동

다음 mv 명령은 단일 파일을 지정된 버킷 및 키로 이동합니다.

```
aws s3 mv test.txt s3://mybucket/test2.txt
```

출력:

```
move: test.txt to s3://mybucket/test2.txt
```

예제 2: 객체를 지정된 버킷 및 키로 이동

다음 mv 명령은 단일 s3 객체를 지정된 버킷 및 키로 이동합니다.

```
aws s3 mv s3://mybucket/test.txt s3://mybucket/test2.txt
```

출력:

```
move: s3://mybucket/test.txt to s3://mybucket/test2.txt
```

예제 3: S3 객체를 로컬 디렉터리로 이동

다음 mv 명령은 단일 객체를 지정된 파일로 로컬로 이동합니다.

```
aws s3 mv s3://mybucket/test.txt test2.txt
```

출력:

```
move: s3://mybucket/test.txt to test2.txt
```

예제 4: 원래 이름이 인 객체를 지정된 버킷으로 이동

다음 mv 명령은 원래 이름을 유지하면서 단일 객체를 지정된 버킷으로 이동합니다.

```
aws s3 mv s3://mybucket/test.txt s3://mybucket2/
```

출력:

```
move: s3://mybucket/test.txt to s3://mybucket2/test.txt
```

예제 5: 버킷의 모든 객체 및 접두사를 로컬 디렉터리로 이동

파라미터 와 함께 전달되면 다음 mv 명령--recursive은 지정된 접두사 및 버킷 아래의 모든 객체를 지정된 디렉터리로 재귀적으로 이동합니다. 이 예제에서는 버킷에 객체 test1.txt 및 mybucket가 있습니다test2.txt.

```
aws s3 mv s3://mybucket . \
```

```
    --recursive
```

출력:

```
move: s3://mybucket/test1.txt to test1.txt
move: s3://mybucket/test2.txt to test2.txt
```

예제 6: ``.jpg`` 파일을 제외한 버킷의 모든 객체 및 접두사를 로컬 디렉터리로 이동

파라미터 와 함께 전달되면 `--recursive`다음 `mv` 명령은 지정된 디렉터리의 모든 파일을 `--exclude` 파라미터를 사용하여 일부 파일을 제외하면서 지정된 버킷 및 접두사로 반복적으로 이동합니다. 이 예제에서는 디렉터리에 파일 `test1.txt` 및 `myDir`가 있습니다`test2.jpg`.

```
aws s3 mv myDir s3://mybucket/ \
    --recursive \
    --exclude "*.jpg"
```

출력:

```
move: myDir/test1.txt to s3://mybucket2/test1.txt
```

예제 7: 지정된 접두사를 제외한 버킷의 모든 객체 및 접두사를 로컬 디렉터리로 이동

파라미터 와 함께 전달되면 `--recursive`다음 `mv` 명령은 `--exclude` 파라미터를 사용하여 일부 객체를 제외하면서 지정된 버킷 아래의 모든 객체를 다른 버킷으로 재귀적으로 이동합니다. 이 예제에서는 버킷에 객체 `test1.txt` 및 `mybucket`가 있습니다`another/test1.txt`.

```
aws s3 mv s3://mybucket/ s3://mybucket2/ \
    --recursive \
    --exclude "mybucket/another/*"
```

출력:

```
move: s3://mybucket/test1.txt to s3://mybucket2/test1.txt
```

예제 8: 객체를 지정된 버킷으로 이동하고 ACL

다음 `mv` 명령은 를 로 설정하는 동안 단일 객체를 지정된 버킷 및 키ACL로 이동합니다`public-read-write`.

```
aws s3 mv s3://mybucket/test.txt s3://mybucket/test2.txt \
    --acl public-read-write
```

출력:

```
move: s3://mybucket/test.txt to s3://mybucket/test2.txt
```

예제 9: 로컬 파일을 지정된 버킷으로 이동하고 권한 부여

다음 mv 명령은 --grants 옵션을 사용하여 모든 사용자에게 읽기 액세스 권한을 부여하고 이메일 주소로 식별된 특정 사용자에게 전체 제어를 부여하는 방법을 보여줍니다.

```
aws s3 mv file.txt s3://mybucket/ \
    --grants read=uri=http://acs.amazonaws.com/groups/global/
AllUsers full=emailaddress=user@example.com
```

출력:

```
move: file.txt to s3://mybucket/file.txt
```

예제 10: 파일을 S3 액세스 포인트로 이동

다음 mv 명령은 라는 단일 파일을 라는 키myaccesspoint에 라는 액세스 포인트mydoc.txt로 이동합니다mykey.

```
aws s3 mv mydoc.txt s3://arn:aws:s3:us-west-2:123456789012:accesspoint/
myaccesspoint/mykey
```

출력:

```
move: mydoc.txt to s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/
mykey
```

- API 자세한 내용은 AWS CLI 명령 참조의 Mv를 참조하세요.

### presign

다음 코드 예시에서는 presign을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: S3 버킷의 객체에 연결하는 기본 1시간 수명URL으로 미리 서명된 을 생성하려면

다음 presign 명령은 1시간 동안 유효한 지정된 버킷 및 키URL에 대해 미리 서명된 를 생성합니다.

```
aws s3 presign s3://DOC-EXAMPLE-BUCKET/test2.txt
```

출력:

```
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
```

예제 2: S3 버킷의 객체에 연결하는 사용자 지정 수명 URL 주기로 미리 서명된 을 생성하려면

다음 presign 명령은 1주일 동안 유효한 지정된 버킷 및 키URL에 대해 미리 서명된 를 생성합니다.

```
aws s3 presign s3://DOC-EXAMPLE-BUCKET/test2.txt \
    --expires-in 604800
```

출력:

```
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
```

자세한 내용은 S3 개발자 안내서의 다른 사람과 객체 공유를 참조하세요.

- API 자세한 내용은 AWS CLI 명령 참조의 Presign을 참조하세요.

### put-bucket-accelerate-configuration

다음 코드 예시에서는 put-bucket-accelerate-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

버킷의 가속화 구성을 설정하려면

다음 put-bucket-accelerate-configuration 예시는 지정된 버킷에 대한 가속화 구성을 활성화합니다.

```
aws s3api put-bucket-accelerate-configuration \
    --bucket my-bucket \
    --accelerate-configuration Status=Enabled
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조PutBucketAccelerateConfiguration의 섹션을 참조하세요. AWS CLI

### put-bucket-acl

다음 코드 예시에서는 put-bucket-acl을 사용하는 방법을 보여 줍니다.

AWS CLI

이 예제에서는 두 명의 AWS 사용자(user1@example.com 및 user2@example.com)full control에게 권한을 부여하고 모든 사용자에게 read 권한을 부여합니다.

```
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
```

사용자 지정에 대한 자세한 내용은 http://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketPUTacl.html을 참조하세요ACLs(와 같은 s3api ACL 명령은 동일한 단축 인수 표기법을 put-bucket-acl사용합니다).

- 자세한 API 내용은 명령 참조PutBucketAcl의 섹션을 참조하세요. AWS CLI

### put-bucket-analytics-configuration

다음 코드 예시에서는 put-bucket-analytics-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

버킷에 대한 분석 구성을 설정하려면

다음 put-bucket-analytics-configuration 예제에서는 지정된 버킷에 대한 분석을 구성합니다.

```
aws s3api put-bucket-analytics-configuration \
    --bucket my-bucket --id 1 \
    --analytics-configuration '{"Id": "1","StorageClassAnalysis": {}}'
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조PutBucketAnalyticsConfiguration의 섹션을 참조하세요. AWS CLI

### put-bucket-cors

다음 코드 예시에서는 put-bucket-cors을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 예시에서는 www.example.com의 PUT, POST, 및 DELETE 요청을 활성화하고 모든 도메인의 GET 요청을 활성화합니다.

```
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
    }
  ]
}
```

- 자세한 API 내용은 명령 참조PutBucketCors의 섹션을 참조하세요. AWS CLI

### put-bucket-encryption

다음 코드 예시에서는 put-bucket-encryption을 사용하는 방법을 보여 줍니다.

AWS CLI

버킷의 서버 측 암호화를 설정하려면

다음 put-bucket-encryption 예제에서는 AES256 암호화를 지정된 버킷의 기본값으로 설정합니다.

```
aws s3api put-bucket-encryption \
    --bucket my-bucket \
    --server-side-encryption-configuration '{"Rules":
 [{"ApplyServerSideEncryptionByDefault": {"SSEAlgorithm": "AES256"}}]}'
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조PutBucketEncryption의 섹션을 참조하세요. AWS CLI

### put-bucket-intelligent-tiering-configuration

다음 코드 예시에서는 put-bucket-intelligent-tiering-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

버킷에서 S3 Intelligent-Tiering 구성을 업데이트하려면

다음 put-bucket-intelligent-tiering-configuration 예제에서는 버킷 ExampleConfig에서 라는 S3 Intelligent-Tiering 구성을 업데이트합니다. 구성은 접두사 이미지에서 액세스하지 않은 객체를 90일 후에 아카이브 액세스로, 180일 후에 딥 아카이브 액세스로 전환합니다.

```
aws s3api put-bucket-intelligent-tiering-configuration \
    --bucket DOC-EXAMPLE-BUCKET \
    --id "ExampleConfig" \
    --intelligent-tiering-configuration file://intelligent-tiering-
configuration.json
```

intelligent-tiering-configuration.json의 콘텐츠:

```
{
```

```
    "Id": "ExampleConfig",
    "Status": "Enabled",
    "Filter": {
        "Prefix": "images"
        },
    "Tierings": [
        {
            "Days": 90,
            "AccessTier": "ARCHIVE_ACCESS"
        },
        {
            "Days": 180,
            "AccessTier": "DEEP_ARCHIVE_ACCESS"
        }
    ]
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon S3 사용 설명서의 기존 버킷에서 객체 소유권 설정을 참조하세요.

- 자세한 API 내용은 명령 참조PutBucketIntelligentTieringConfiguration의 섹션을 참조하세요. AWS CLI

### put-bucket-inventory-configuration

다음 코드 예시에서는 put-bucket-inventory-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 버킷에 대한 인벤토리 구성을 설정하려면

다음 put-bucket-inventory-configuration 예제에서는 버킷 에 대한 주간 ORC형식 인벤토리 보고서를 설정합니다my-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" }}'
```

이 명령은 출력을 생성하지 않습니다.

예제 2: 버킷에 대한 인벤토리 구성을 설정하려면

다음 put-bucket-inventory-configuration 예제에서는 버킷 에 대한 일일 CSV형식 인벤토리 보고서를 설정합니다my-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" }}'
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조PutBucketInventoryConfiguration의 섹션을 참조하세요. AWS CLI

### put-bucket-lifecycle-configuration

다음 코드 예시에서는 put-bucket-lifecycle-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 my-bucket이라는 버킷에 수명 주기 구성을 적용합니다.

```
aws s3api put-bucket-lifecycle-configuration --bucket my-bucket --lifecycle-
configuration  file://lifecycle.json
```

파일은 현재 폴더의 JSON 문서lifecycle.json로, 두 가지 규칙을 지정합니다.

```
{
    "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"
        }
    ]
}
```

첫 번째 규칙은 rotated 접두사가 있는 파일을 지정된 날짜에 Glacier로 옮깁니다. 두 번째 규칙은 이전 객체 버전이 더 이상 최신 버전이 아닌 경우 Glacier로 옮깁니다. 허용되는 타임스탬프 형식에 대한 자세한 내용은 AWS CLI 사용 설명서의 파라미터 값 지정을 참조하세요.

- 자세한 API 내용은 명령 참조PutBucketLifecycleConfiguration의 섹션을 참조하세요. AWS CLI

### put-bucket-lifecycle

다음 코드 예시에서는 put-bucket-lifecycle을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 버킷 에 수명 주기 구성을 적용합니다my-bucket.

```
aws s3api put-bucket-lifecycle --bucket my-bucket --lifecycle-configuration file://
lifecycle.json
```

파일은 현재 폴더의 JSON 문서lifecycle.json로, 두 가지 규칙을 지정합니다.

```
{
  "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"
    }
  ]
}
```

첫 번째 규칙은 60일 후에 Amazon Glacier로 파일을 이동합니다. 두 번째 규칙은 지정된 날짜에 Amazon S3에서 파일을 삭제합니다. 허용되는 타임스탬프 형식에 대한 자세한 내용은 AWS CLI 사용 설명서의 파라미터 값 지정을 참조하세요.

위 예제의 각 규칙은 적용되는 정책(`Transition` 또는 `Expiration`) 및 파일 접두사(폴더 이름)를 지정합니다. 빈 접두사를 지정하여 전체 버킷에 적용되는 규칙을 생성할 수도 있습니다.

```
{
  "Rules": [
    {
      "ID": "Move to Glacier after sixty days (all objects in bucket)",
      "Prefix": "",
      "Status": "Enabled",
      "Transition": {
        "Days": 60,
        "StorageClass": "GLACIER"
      }
    }
  ]
}
```

- 자세한 API 내용은 명령 참조PutBucketLifecycle의 섹션을 참조하세요. AWS CLI

### `put-bucket-logging`

다음 코드 예시에서는 `put-bucket-logging`을 사용하는 방법을 보여 줍니다.

AWS CLI

예시 1: 버킷 정책 로깅을 설정하려면

다음 put-bucket-logging 예제에서는 에 대한 로깅 정책을 설정합니다MyBucket. 먼저 put-bucket-policy 명령을 사용하여 버킷 정책에서 로깅 서비스 보안 주체 권한을 부여합니다.

```
aws s3api put-bucket-policy \
    --bucket MyBucket \
    --policy file://policy.json
```

policy.json의 콘텐츠:

```
{
    "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"}
            }
        }
    ]
}
```

로깅 정책을 적용하려면 put-bucket-logging을 사용합니다.

```
aws s3api put-bucket-logging \
    --bucket MyBucket \
    --bucket-logging-status file://logging.json
```

logging.json의 콘텐츠:

```
{
     "LoggingEnabled": {
         "TargetBucket": "MyBucket",
         "TargetPrefix": "Logs/"
```

```
    }
}
```

로깅 서비스 보안 주체에 `s3:PutObject` 권한을 부여하려면 `put-bucket-policy` 명령이 필요합니다.

자세한 내용은 Amazon S3 사용 설명서의 Amazon S3 서버 액세스 로깅을 참조하세요.

예시 2: 단일 사용자에게만 액세스 로깅에 대한 버킷 정책을 설정하려면

다음 `put-bucket-logging` 예제에서는 에 대한 로깅 정책을 설정합니다MyBucket. AWS 사용자 bob@example.com은 로그 파일을 완전히 제어할 수 있으며 다른 사람은 액세스할 수 없습니다. 먼저 `put-bucket-acl`을 사용하여 S3 권한을 부여합니다.

```
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
```

그런 다음 `put-bucket-logging`을 사용하여 로깅 정책을 적용합니다.

```
aws s3api put-bucket-logging \
    --bucket MyBucket \
    --bucket-logging-status file://logging.json
```

`logging.json`의 콘텐츠:

```
{
    "LoggingEnabled": {
        "TargetBucket": "MyBucket",
        "TargetPrefix": "MyBucketLogs/",
        "TargetGrants": [
            {
                "Grantee": {
                    "Type": "AmazonCustomerByEmail",
                    "EmailAddress": "bob@example.com"
                },
                "Permission": "FULL_CONTROL"
            }
        ]
    }
}
```

S3의 로그 전달 시스템에 필수 권한(write 및 read-acp 권한)을 부여하려면 `put-bucket-acl` 명령이 필요합니다.

자세한 내용은 Amazon S3 개발자 안내서의 Amazon S3 서버 액세스 로깅을 참조하세요.

- 자세한 API 내용은 명령 참조PutBucketLogging의 섹션을 참조하세요. AWS CLI

### put-bucket-metrics-configuration

다음 코드 예시에서는 `put-bucket-metrics-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

버킷에 대한 지표 구성을 설정하려면

다음 `put-bucket-metrics-configuration` 예제에서는 지정된 버킷에 대해 ID 123으로 지표 구성을 설정합니다.

```
aws s3api put-bucket-metrics-configuration \
    --bucket my-bucket \
    --id 123 \
    --metrics-configuration '{"Id": "123", "Filter": {"Prefix": "logs"}}'
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조PutBucketMetricsConfiguration의 섹션을 참조하세요. AWS CLI

### put-bucket-notification-configuration

다음 코드 예시에서는 `put-bucket-notification-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

버킷에 지정된 알림을 활성화하려면

다음 `put-bucket-notification-configuration` 예시에서는 `my-bucket`이라는 버킷에 알림 구성을 적용합니다. 파일은 모니터링할 SNS 주제와 이벤트 유형을 지정하는 현재 폴더의 JSON 문서`notification.json`입니다.

```
aws s3api put-bucket-notification-configuration \
```

```
    --bucket my-bucket \
    --notification-configuration file://notification.json
```

notification.json의 콘텐츠:

```
{
    "TopicConfigurations": [
        {
            "TopicArn": "arn:aws:sns:us-west-2:123456789012:s3-notification-topic",
            "Events": [
                "s3:ObjectCreated:*"
            ]
        }
    ]
}
```

SNS 주제에는 Amazon S3가 게시할 수 있는 IAM 정책이 연결되어 있어야 합니다.

```
{
    "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"
                }
            }
        }
    ]
}
```

- 자세한 API 내용은 명령 참조PutBucketNotificationConfiguration의 섹션을 참조하세요. AWS CLI

### put-bucket-notification

다음 코드 예시에서는 put-bucket-notification을 사용하는 방법을 보여 줍니다.

AWS CLI

my-bucket이라는 버킷에 알림 구성을 적용합니다.

```
aws s3api put-bucket-notification --bucket my-bucket --notification-
configuration file://notification.json
```

파일은 모니터링할 SNS 주제와 이벤트 유형을 지정하는 현재 폴더의 JSON 문서notification.json입니다.

```
{
  "TopicConfiguration": {
    "Event": "s3:ObjectCreated:*",
    "Topic": "arn:aws:sns:us-west-2:123456789012:s3-notification-topic"
  }
}
```

SNS 주제에는 Amazon S3가 게시할 수 있도록 허용하는 IAM 정책이 연결되어 있어야 합니다.

```
{
 "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"
    }
   }
```

```
  }
 ]
}
```

- 자세한 API 내용은 명령 참조PutBucketNotification의 섹션을 참조하세요. AWS CLI

### put-bucket-ownership-controls

다음 코드 예시에서는 put-bucket-ownership-controls을 사용하는 방법을 보여 줍니다.

AWS CLI

버킷의 버킷 소유권 설정을 업데이트하려면

다음 put-bucket-ownership-controls 예제에서는 버킷의 버킷 소유권 설정을 업데이트합니다.

```
aws s3api put-bucket-ownership-controls \
    --bucket DOC-EXAMPLE-BUCKET \
    --ownership-controls="Rules=[{ObjectOwnership=BucketOwnerEnforced}]"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon S3 사용 설명서의 기존 버킷에서 객체 소유권 설정을 참조하세요.

- 자세한 API 내용은 명령 참조PutBucketOwnershipControls의 섹션을 참조하세요. AWS CLI

### put-bucket-policy

다음 코드 예시에서는 put-bucket-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

이 예제에서는 모든 사용자가 의 객체를 MyBucket 제외한 의 객체를 검색할 수 있습니다MySecretFolder. 또한 AWS 계정의 루트 사용자에게 1234-5678-9012다음과 같은 put delete 권한을 부여합니다.

```
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/*"
      }

   ]
}
```

- 자세한 API 내용은 명령 참조PutBucketPolicy의 섹션을 참조하세요. AWS CLI

### put-bucket-replication

다음 코드 예시에서는 put-bucket-replication을 사용하는 방법을 보여 줍니다.

AWS CLI

S3 버킷의 복제를 구성하려면

다음 put-bucket-replication 예시는 지정된 S3 버킷에 복제 구성을 적용합니다.

```
aws s3api put-bucket-replication \
    --bucket AWSDOC-EXAMPLE-BUCKET1 \
    --replication-configuration file://replication.json
```

replication.json의 콘텐츠:

```
{
    "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"
            }
        }
    ]
}
```

대상 버킷에 버전 관리가 활성화되어 있어야 합니다. 지정된 역할에는 대상 버킷에 쓰기 위한 권한이 있어야 하며 Amazon S3가 역할을 맡도록 허용하는 신뢰 관계가 있어야 합니다.

예시 역할 권한 정책:

```
{
    "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/*"
        }
    ]
}
```

예시 신뢰 관계 정책:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "s3.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Simple Storage Service 콘솔 사용 설명서의 주제 제목을 참조하세요.

- 자세한 API 내용은 명령 참조PutBucketReplication의 섹션을 참조하세요. AWS CLI

### `put-bucket-request-payment`

다음 코드 예시에서는 `put-bucket-request-payment`을 사용하는 방법을 보여 줍니다.

AWS CLI

예시 1: 버킷의 ``요청자 지불`` 구성을 활성화하려면

다음 put-bucket-request-payment 예시는 지정된 버킷의 requester pays를 활성화합니다.

```
aws s3api put-bucket-request-payment \
    --bucket my-bucket \
    --request-payment-configuration '{"Payer":"Requester"}'
```

이 명령은 출력을 생성하지 않습니다.

예시 2: 버킷의 ``요청자 지불`` 구성을 비활성화하려면

다음 put-bucket-request-payment 예시는 지정된 버킷의 requester pays를 비활성화합니다.

```
aws s3api put-bucket-request-payment \
    --bucket my-bucket \
    --request-payment-configuration '{"Payer":"BucketOwner"}'
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조PutBucketRequestPayment의 섹션을 참조하세요. AWS CLI

### put-bucket-tagging

다음 코드 예시에서는 put-bucket-tagging을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 my-bucket이라는 버킷에 태그 지정 구성을 적용합니다.

```
aws s3api put-bucket-tagging --bucket my-bucket --tagging file://tagging.json
```

파일은 태그를 지정하는 현재 폴더의 JSON 문서tagging.json입니다.

```
{
   "TagSet": [
     {
       "Key": "organization",
       "Value": "marketing"
     }
   ]
```

```
}
```

또는 명령줄에서 태그 지정 구성을 my-bucket에 직접 적용할 수도 있습니다.

```
aws s3api put-bucket-tagging --bucket my-bucket --tagging
 'TagSet=[{Key=organization,Value=marketing}]'
```

- 자세한 API 내용은 명령 참조PutBucketTagging의 섹션을 참조하세요. AWS CLI

### put-bucket-versioning

다음 코드 예시에서는 put-bucket-versioning을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 이름이 my-bucket인 버킷의 버전 관리를 활성화합니다.

```
aws s3api put-bucket-versioning --bucket my-bucket --versioning-
configuration Status=Enabled
```

다음 명령은 버전 관리를 활성화하고 mfa 코드를 사용합니다.

```
aws s3api put-bucket-versioning --bucket my-bucket --versioning-
configuration Status=Enabled --mfa "SERIAL 123456"
```

- 자세한 API 내용은 명령 참조PutBucketVersioning의 섹션을 참조하세요. AWS CLI

### put-bucket-website

다음 코드 예시에서는 put-bucket-website을 사용하는 방법을 보여 줍니다.

AWS CLI

my-bucket이라는 버킷에 정적 웹 사이트 구성을 적용합니다.

```
aws s3api put-bucket-website --bucket my-bucket --website-configuration file://
website.json
```

파일은 웹 사이트의 인덱스 및 오류 페이지를 지정하는 현재 폴더의 JSON 문서website.json입니다.

```
{
    "IndexDocument": {
        "Suffix": "index.html"
    },
    "ErrorDocument": {
        "Key": "error.html"
    }
}
```

- 자세한 API 내용은 명령 참조PutBucketWebsite의 섹션을 참조하세요. AWS CLI

### put-object-acl

다음 코드 예시에서는 put-object-acl을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 두 명의 AWS 사용자(user1@example.com 및 user2@example.com)full control에게 권한을 부여하고 모든 사용자에게 read 권한을 부여합니다.

```
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
```

사용자 지정에 대한 자세한 내용은 http://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketPUTacl.html을 참조하세요ACLs(와 같은 s3api ACL 명령은 동일한 단축 인수 표기법을 put-object-acl사용합니다).

- 자세한 API 내용은 명령 참조PutObjectAcl의 섹션을 참조하세요. AWS CLI

### put-object-legal-hold

다음 코드 예시에서는 put-object-legal-hold을 사용하는 방법을 보여 줍니다.

AWS CLI

객체에 법적 보존을 적용하는 방법

다음 put-object-legal-hold 예시에서는 doc1.rtf 객체에 법적 보존을 설정합니다.

```
aws s3api put-object-legal-hold \
    --bucket my-bucket-with-object-lock \
```

```
    --key doc1.rtf \
    --legal-hold Status=ON
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조PutObjectLegalHold의 섹션을 참조하세요. AWS CLI

### put-object-lock-configuration

다음 코드 예시에서는 put-object-lock-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

버킷에 객체 잠금 구성을 설정하는 방법

다음 put-object-lock-configuration 예시에서는 지정된 버킷에 50일 객체 잠금을 설정합니다.

```
aws s3api put-object-lock-configuration \
    --bucket my-bucket-with-object-lock \
    --object-lock-configuration '{ "ObjectLockEnabled": "Enabled", "Rule":
 { "DefaultRetention": { "Mode": "COMPLIANCE", "Days": 50 }}}'
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조PutObjectLockConfiguration의 섹션을 참조하세요. AWS CLI

### put-object-retention

다음 코드 예시에서는 put-object-retention을 사용하는 방법을 보여 줍니다.

AWS CLI

객체의 객체 보존 구성을 설정하는 방법

다음 put-object-retention 예시에서는 지정된 객체에 2025-01-01까지 객체 보존 구성을 설정합니다.

```
aws s3api put-object-retention \
    --bucket my-bucket-with-object-lock \
    --key doc1.rtf \
    --retention '{ "Mode": "GOVERNANCE", "RetainUntilDate": "2025-01-01T00:00:00" }'
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조PutObjectRetention의 섹션을 참조하세요. AWS CLI

### put-object-tagging

다음 코드 예시에서는 put-object-tagging을 사용하는 방법을 보여 줍니다.

AWS CLI

객체에 태그를 설정하려면

다음 put-object-tagging 예제에서는 키designation와 confidential 지정된 객체의 값을 사용하여 태그를 설정합니다.

```
aws s3api put-object-tagging \
    --bucket my-bucket \
    --key doc1.rtf \
    --tagging '{"TagSet": [{ "Key": "designation", "Value": "confidential" }]}'
```

이 명령은 출력을 생성하지 않습니다.

다음 put-object-tagging 예제에서는 지정된 객체에 여러 태그 세트를 설정합니다.

```
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" } ]}'
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조PutObjectTagging의 섹션을 참조하세요. AWS CLI

### put-object

다음 코드 예시에서는 put-object을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 예시에서는 put-object 명령을 사용하여 Amazon S3에 객체를 업로드합니다.

```
aws s3api put-object --bucket text-content --key dir-1/my_images.tar.bz2 --
body my_images.tar.bz2
```

다음 예시에서는 비디오 파일의 업로드를 보여줍니다(비디오 파일은 Windows 파일 시스템 구문을 사용하여 지정됨).

```
aws s3api put-object --bucket text-content --key dir-1/big-video-file.mp4 --body e:
\media\videos\f-sharp-3-data-services.mp4
```

객체 업로드에 대한 자세한 내용은 Amazon S3 개발자 안내서의 객체 업로드를 참조하세요.

- 자세한 API 내용은 명령 참조PutObject의 섹션을 참조하세요. AWS CLI

### put-public-access-block

다음 코드 예시에서는 put-public-access-block을 사용하는 방법을 보여 줍니다.

AWS CLI

버킷에 대한 퍼블릭 액세스 차단 구성을 설정하려면

다음 put-public-access-block 예제에서는 지정된 버킷에 대한 제한적인 블록 퍼블릭 액세스 구성을 설정합니다.

```
aws s3api put-public-access-block \
    --bucket my-bucket \
    --public-access-block-
configuration "BlockPublicAcls=true,IgnorePublicAcls=true,BlockPublicPolicy=true,RestrictPub
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조PutPublicAccessBlock의 섹션을 참조하세요. AWS CLI

### rb

다음 코드 예시에서는 rb을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 버킷 삭제

다음 rb 명령은 버킷을 제거합니다. 이 예제에서 사용자의 버킷은 입니다mybucket. 버킷을 제거하려면 비어 있어야 합니다.

```
aws s3 rb s3://mybucket
```

출력:

```
remove_bucket: mybucket
```

예제 2: 버킷 강제 삭제

다음 rb 명령은 --force 파라미터를 사용하여 먼저 버킷의 모든 객체를 제거한 다음 버킷 자체를 제거합니다. 이 예제에서 사용자의 버킷은 mybucket 이고 의 객체mybucket는 test1.txt 및 입니다test2.txt.

```
aws s3 rb s3://mybucket \
    --force
```

출력:

```
delete: s3://mybucket/test1.txt
delete: s3://mybucket/test2.txt
remove_bucket: mybucket
```

- API 자세한 내용은 AWS CLI 명령 참조의 Rb를 참조하세요.

### restore-object

다음 코드 예시에서는 restore-object을 사용하는 방법을 보여 줍니다.

AWS CLI

객체에 대한 복원 요청을 생성하는 방법

다음 restore-object 예시에서는 my-glacier-bucket 버킷의 지정된 Amazon S3 Glacier 객체를 10일 동안 복원합니다.

```
aws s3api restore-object \
    --bucket my-glacier-bucket \
    --key doc1.rtf \
```

```
    --restore-request Days=10
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조RestoreObject의 섹션을 참조하세요. AWS CLI

### rm

다음 코드 예시에서는 rm을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: S3 객체 삭제

다음 rm 명령은 단일 s3 객체를 삭제합니다.

```
aws s3 rm s3://mybucket/test2.txt
```

출력:

```
delete: s3://mybucket/test2.txt
```

예제 2: 버킷의 모든 콘텐츠 삭제

다음 rm 명령은 파라미터 와 함께 전달될 때 지정된 버킷 및 접두사 아래의 모든 객체를 반복적으로 삭제합니다--recursive. 이 예제에서는 버킷에 객체 test1.txt 및 가 mybucket 포함되어 있습니다test2.txt.

```
aws s3 rm s3://mybucket \
    --recursive
```

출력:

```
delete: s3://mybucket/test1.txt
delete: s3://mybucket/test2.txt
```

예제 3: ``.jpg` 파일을 제외한 버킷의 모든 콘텐츠 삭제

다음 rm 명령은 파라미터를 사용하여 일부 객체를 제외--recursive하면서 --exclude 파라미터와 함께 전달될 때 지정된 버킷 및 접두사 아래의 모든 객체를 반복적으로 삭제합니다. 이 예제에서는 버킷에 객체 test1.txt 및 mybucket가 있습니다test2.jpg.

```
aws s3 rm s3://mybucket/ \
    --recursive \
    --exclude "*.jpg"
```

출력:

```
delete: s3://mybucket/test1.txt
```

예제 4: 지정된 접두사 아래의 객체를 제외한 버킷의 모든 콘텐츠 삭제

다음 rm 명령은 파라미터를 사용하여 특정 접두사 아래의 모든 객체를 제외--recursive하면서 --exclude 파라미터와 함께 전달될 때 지정된 버킷 및 접두사 아래의 모든 객체를 반복적으로 삭제합니다. 이 예제에서는 버킷에 객체 test1.txt 및 mybucket가 있습니다another/test.txt.

```
aws s3 rm s3://mybucket/ \
    --recursive \
    --exclude "another/*"
```

출력:

```
delete: s3://mybucket/test1.txt
```

예제 5: S3 액세스 포인트에서 객체 삭제

다음 rm 명령은 액세스 포인트(mykey)에서 단일 객체()를 삭제합니다myaccesspoint. :: 다음 rm 명령은 액세스 포인트(mykey)에서 단일 객체()를 삭제합니다myaccesspoint.

```
aws s3 rm s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/mykey
```

출력:

```
delete: s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/mykey
```

- API 자세한 내용은 AWS CLI 명령 참조의 Rm을 참조하세요.

### select-object-content

다음 코드 예시에서는 select-object-content을 사용하는 방법을 보여 줍니다.

AWS CLI

SQL 문을 기반으로 Amazon S3 객체의 콘텐츠를 필터링하려면

다음 `select-object-content` 예제에서는 지정된 SQL 문`my-data-file.csv`으로 객체를 필터링하고 출력을 파일로 전송합니다.

```
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"
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조SelectObjectContent의 섹션을 참조하세요. AWS CLI

**sync**

다음 코드 예시에서는 `sync`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 모든 로컬 객체를 지정된 버킷에 동기화

다음 `sync` 명령은 로컬 파일을 S3에 업로드하여 로컬 디렉터리의 객체를 지정된 접두사 및 버킷과 동기화합니다. 로컬 파일의 크기가 S3 객체의 크기와 다르거나, 로컬 파일의 마지막 수정 시간이 S3 객체의 마지막 수정 시간보다 빠르거나, 지정된 버킷 및 접두사 아래에 로컬 파일이 없는 경우 로컬 파일을 업로드해야 합니다. 이 예제에서는 사용자가 버킷을 로컬 현재 디렉터리`mybucket`와 동기화합니다. 로컬 현재 디렉터리에는 `test.txt` 및 파일이 포함되어 있습니다`test2.txt`. 버킷에 객체`mybucket`가 없습니다.

```
aws s3 sync . s3://mybucket
```

출력:

```
upload: test.txt to s3://mybucket/test.txt
upload: test2.txt to s3://mybucket/test2.txt
```

예제 2: 지정된 S3 버킷의 모든 S3 객체를 다른 버킷과 동기화

다음 sync 명령은 S3 객체를 복사하여 지정된 접두사 및 버킷 아래의 객체를 다른 지정된 접두사 및 버킷 아래의 객체와 동기화합니다. 두 S3 객체의 크기가 S3 다르거나, 소스의 마지막 수정 시간이 대상의 마지막 수정 시간보다 빠르거나, 지정된 버킷 및 접두사 대상 아래에 S3 객체가 없는 경우 S3 객체를 복사해야 합니다.

이 예제에서는 사용자가 버킷을 버킷 mybucket에 동기화합니다mybucket2. 버킷에는 객체 test.txt 및 가 mybucket 포함되어 있습니다test2.txt. 버킷에는 객체mybucket2가 없습니다.

```
aws s3 sync s3://mybucket s3://mybucket2
```

출력:

```
copy: s3://mybucket/test.txt to s3://mybucket2/test.txt
copy: s3://mybucket/test2.txt to s3://mybucket2/test2.txt
```

예제 3: 지정된 S3 버킷의 모든 S3 객체를 로컬 디렉터리로 동기화

다음 sync 명령은 S3 객체를 다운로드하여 지정된 S3 버킷의 파일을 로컬 디렉터리로 동기화합니다. S3 객체의 크기가 로컬 파일의 크기와 다르거나, S3 S3 객체의 마지막 수정 시간이 로컬 파일의 마지막 수정 시간보다 빠르거나, S3 객체가 로컬 디렉터리에 없는 경우 S3 객체를 다운로드해야 합니다. 객체를 S3에서 다운로드하면 로컬 파일의 마지막 수정 시간이 S3 객체의 마지막 수정 시간으로 변경됩니다. 이 예제에서는 사용자가 버킷을 현재 로컬 디렉터리mybucket와 동기화합니다. 버킷에는 객체 test.txt 및 가 mybucket 포함되어 있습니다test2.txt. 현재 로컬 디렉터리에는 파일이 없습니다.

```
aws s3 sync s3://mybucket .
```

출력:

```
download: s3://mybucket/test.txt to test.txt
download: s3://mybucket/test2.txt to test2.txt
```

예제 4: 모든 로컬 객체를 지정된 버킷에 동기화하고 일치하지 않는 모든 파일을 삭제합니다.

다음 sync 명령은 지정된 접두사 및 버킷 아래의 객체를 로컬 파일을 S3에 업로드하여 로컬 디렉터리의 파일에 동기화합니다. --delete 파라미터로 인해 지정된 접두사 및 버킷 아래에 있지만

로컬 디렉터리에는 없는 모든 파일이 삭제됩니다. 이 예제에서는 사용자가 버킷을 로컬 현재 디렉터리mybucket와 동기화합니다. 로컬 현재 디렉터리에는 test.txt 및 파일이 포함되어 있습니다test2.txt. 버킷에는 객체 가 mybucket 포함되어 있습니다test3.txt.

```
aws s3 sync . s3://mybucket \
    --delete
```

출력:

```
upload: test.txt to s3://mybucket/test.txt
upload: test2.txt to s3://mybucket/test2.txt
delete: s3://mybucket/test3.txt
```

예제 5: ``.jpg`` 파일을 제외한 모든 로컬 객체를 지정된 버킷에 동기화

다음 sync 명령은 지정된 접두사 및 버킷 아래의 객체를 로컬 파일을 S3에 업로드하여 로컬 디렉터리의 파일에 동기화합니다. --exclude 파라미터로 인해 S3 및 로컬 모두에 존재하는 패턴과 일치하는 모든 파일은 동기화에서 제외됩니다. 이 예제에서는 사용자가 버킷을 로컬 현재 디렉터리mybucket와 동기화합니다. 로컬 현재 디렉터리에는 test.jpg 및 파일이 포함되어 있습니다test2.txt. 버킷에는 로컬 test.jpg과 다른 크기의 객체가 mybucket 포함되어 있습니다test.jpg.

```
aws s3 sync . s3://mybucket \
    --exclude "*.jpg"
```

출력:

```
upload: test2.txt to s3://mybucket/test2.txt
```

예제 6: ``.jpg`` 파일을 제외한 모든 로컬 객체를 지정된 버킷에 동기화

다음 sync 명령은 S3 객체를 다운로드하여 로컬 디렉터리의 파일을 지정된 접두사 및 버킷의 객체와 동기화합니다. 이 예제에서는 --exclude 파라미터 플래그를 사용하여 지정된 디렉터리와 S3 접두사를 sync 명령에서 제외합니다. 이 예제에서는 사용자가 로컬 현재 디렉터리를 버킷 에 동기화합니다mybucket. 로컬 현재 디렉터리에는 test.txt 및 파일이 포함되어 있습니다another/test2.txt. 버킷에는 객체 another/test5.txt 및 가 mybucket 포함됩니다test1.txt.

```
aws s3 sync s3://mybucket/ . \
```

```
    --exclude "*another/*"
```

출력:

```
download: s3://mybucket/test1.txt to test1.txt
```

예제 7: 서로 다른 리전의 버킷 간에 모든 객체 동기화

다음 sync 명령은 서로 다른 리전의 두 버킷 간에 파일을 동기화합니다.

```
aws s3 sync s3://my-us-west-2-bucket s3://my-us-east-1-bucket \
    --source-region us-west-2 \
    --region us-east-1
```

출력:

```
download: s3://my-us-west-2-bucket/test1.txt to s3://my-us-east-1-bucket/test1.txt
```

예제 8: S3 액세스 포인트에 동기화

다음 sync 명령은 현재 디렉터리를 액세스 포인트()와 동기화합니다myaccesspoint.

```
aws s3 sync . s3://arn:aws:s3:us-west-2:123456789012:accesspoint/myaccesspoint/
```

출력:

```
upload: test.txt to s3://arn:aws:s3:us-west-2:123456789012:accesspoint/
myaccesspoint/test.txt
upload: test2.txt to s3://arn:aws:s3:us-west-2:123456789012:accesspoint/
myaccesspoint/test2.txt
```

- API 자세한 내용은 AWS CLI 명령 참조의 동기화를 참조하세요.

### upload-part-copy

다음 코드 예시에서는 upload-part-copy을 사용하는 방법을 보여 줍니다.

AWS CLI

기존 객체의 데이터를 데이터 소스로 복사하여 객체의 일부를 업로드하려면

다음 upload-part-copy 예제에서는 기존 객체의 데이터를 데이터 소스로 복사하여 파트를 업로드합니다.

```
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_RiBy0ggOH6b13pVRY7QjvLlf75iFdJqp_2wztk5hvpUM2SesXgrzbeh
```

출력:

```
{
    "CopyPartResult": {
        "LastModified": "2019-12-13T23:16:03.000Z",
        "ETag": "\"711470fc377698c393d94aed6305e245\""
    }
}
```

- 자세한 API 내용은 명령 참조UploadPartCopy의 섹션을 참조하세요. AWS CLI

### upload-part

다음 코드 예시에서는 upload-part을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 create-multipart-upload 명령으로 시작된 멀티파트 업로드의 첫 번째 파트를 업로드합니다.

```
aws s3api upload-part --bucket my-bucket --key 'multipart/01' --part-number 1 --
body part01 --upload-id
  "dfRtDYU0WWCCcH43C3WFbkRONycyCpTJJvxu2i5GYkZljF.Yxwh6XG7WfS2vC4to6HiV6Yjlx.cph0gtNBtJ8P3UR
```

body 옵션은 업로드할 로컬 파일의 이름 또는 경로를 사용합니다. file:// 접두사는 사용하지 마세요. 최소 파트 크기는 5MB입니다. create-multipart-upload에서 업로드 ID를 반환하며 list-multipart-uploads를 사용하여 검색할 수도 있습니다. 멀티파트 업로드를 생성할 때 버킷과 키가 지정됩니다.

출력:

```
{
    "ETag": "\"e868e0f4719e394144ef36531ee6824c\""
}
```

나중에 사용할 수 있도록 각 부분의 ETag 값을 저장합니다. 멀티파트 업로드를 완료하는 데 필요합니다.

- 자세한 API 내용은 명령 참조UploadPart의 섹션을 참조하세요. AWS CLI

### website

다음 코드 예시에서는 website을 사용하는 방법을 보여 줍니다.

AWS CLI

정적 웹 사이트로 S3 버킷 구성

다음 명령은 정적 웹 사이트my-bucket라는 버킷을 구성합니다. 인덱스 문서 옵션은 방문자my-bucket가 웹 사이트 로 이동할 때 로 이동할 파일을 지정합니다URL. 이 경우 버킷은 us-west-2 리전에 있으므로 사이트는 에 표시됩니다http://my-bucket.s3-website-us-west-2.amazonaws.com.

정적 사이트에 표시되는 버킷의 모든 파일은 방문자가 파일을 열 수 있도록 구성해야 합니다. 파일 권한은 버킷 웹 사이트 구성과 별도로 구성됩니다.

```
aws s3 website s3://my-bucket/ \
    --index-document index.html \
    --error-document error.html
```

Amazon S3에서 정적 웹 사이트를 호스팅하는 방법에 대한 자세한 내용은 Amazon Simple Storage Service 개발자 안내서의 정적 웹 사이트 호스팅을 참조하세요.

- API 자세한 내용은 명령 참조의 웹 사이트를 참조하세요. AWS CLI

## 를 사용한 Amazon S3 컨트롤 예제 AWS CLI

다음 코드 예제에서는 Amazon S3 Control과 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `create-access-point`

다음 코드 예시에서는 `create-access-point`을 사용하는 방법을 보여 줍니다.

AWS CLI

액세스 포인트를 생성하려면

다음 `create-access-point` 예제에서는 계정 123456789012의 버킷`business-records`에 `finance-ap` 대한 라는 액세스 포인트를 생성합니다. 이 예제를 실행하기 전에 액세스 포인트 이름, 버킷 이름 및 계정 번호를 사용 사례에 적합한 값으로 바꿉니다.

```
aws s3control create-access-point \
    --account-id 123456789012 \
    --bucket business-records \
    --name finance-ap
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Simple Storage Service 개발자 안내서의 액세스 포인트 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateAccessPoint의 섹션을 참조하세요. AWS CLI

### `create-job`

다음 코드 예시에서는 `create-job`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon S3 배치 작업 작업을 생성하려면

다음 create-job 예제에서는 객체를 로 태그 지정하기 위한 Amazon S3 배치 작업 작업을 생성합니다confidential` 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
```

출력:

```
{
    "JobId": "93735294-df46-44d5-8638-6356f335324e"
}
```

- 자세한 API 내용은 명령 참조CreateJob의 섹션을 참조하세요. AWS CLI

### delete-access-point-policy

다음 코드 예시에서는 delete-access-point-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

액세스 포인트 정책을 삭제하려면

다음 delete-access-point-policy 예제에서는 계정 123456789012finance-ap에 이름이 지정된 액세스 포인트에서 액세스 포인트 정책을 삭제합니다. 이 예제를 실행하기 전에 액세스 포인트 이름과 계정 번호를 사용 사례에 적합한 값으로 바꿉니다.

```
aws s3control delete-access-point-policy \
```

```
    --account-id 123456789012 \
    --name finance-ap
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Simple Storage Service 개발자 안내서의 Amazon S3 액세스 포인트를 사용한 데이터 액세스 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteAccessPointPolicy의 섹션을 참조하세요. AWS CLI

### delete-access-point

다음 코드 예시에서는 delete-access-point을 사용하는 방법을 보여 줍니다.

AWS CLI

액세스 포인트를 삭제하려면

다음 delete-access-point 예제에서는 계정 123456789012finance-ap에 이름이 지정된 액세스 포인트를 삭제합니다. 이 예제를 실행하기 전에 액세스 포인트 이름과 계정 번호를 사용 사례에 적합한 값으로 바꿉니다.

```
aws s3control delete-access-point \
    --account-id 123456789012 \
    --name finance-ap
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Simple Storage Service 개발자 안내서의 Amazon S3 액세스 포인트를 사용한 데이터 액세스 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteAccessPoint의 섹션을 참조하세요. AWS CLI

### delete-public-access-block

다음 코드 예시에서는 delete-public-access-block을 사용하는 방법을 보여 줍니다.

AWS CLI

계정에 대한 퍼블릭 액세스 차단 설정을 삭제하려면

다음 delete-public-access-block 예제에서는 지정된 계정에 대한 퍼블릭 액세스 차단 설정을 삭제합니다.

```
aws s3control delete-public-access-block \
    --account-id 123456789012
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeletePublicAccessBlock의 섹션을 참조하세요. AWS CLI

### describe-job

다음 코드 예시에서는 describe-job을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon S3 배치 작업 작업 설명

다음은 지정된 배치 작업 작업에 대한 구성 파라미터 및 상태를 describe-job 제공합니다.

```
aws s3control describe-job \
    --account-id 123456789012 \
    --job-id 93735294-df46-44d5-8638-6356f335324e
```

출력:

```
{
    "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"
    }
}
```

- 자세한 API 내용은 명령 참조DescribeJob의 섹션을 참조하세요. AWS CLI

### `get-access-point-policy-status`

다음 코드 예시에서는 `get-access-point-policy-status`을 사용하는 방법을 보여 줍니다.

AWS CLI

액세스 포인트 정책 상태를 검색하려면

다음 `get-access-point-policy-status` 예제에서는 계정 123456789012finance-ap에 이름이 지정된 액세스 포인트의 액세스 포인트 정책 상태를 검색합니다. 액세스 포인트 정책 상태는

액세스 포인트의 정책이 퍼블릭 액세스를 허용하는지 여부를 나타냅니다. 이 예제를 실행하기 전에 액세스 포인트 이름과 계정 번호를 사용 사례에 적합한 값으로 바꿉니다.

```
aws s3control get-access-point-policy-status \
    --account-id 123456789012 \
    --name finance-ap
```

출력:

```
{
    "PolicyStatus": {
        "IsPublic": false
    }
}
```

액세스 포인트 정책이 퍼블릭으로 간주되는 시기에 대한 자세한 내용은 Amazon Simple Storage Service 개발자 안내서의 '공개'의 의미를 참조하세요.

- 자세한 API 내용은 명령 참조GetAccessPointPolicyStatus의 섹션을 참조하세요. AWS CLI

### get-access-point-policy

다음 코드 예시에서는 get-access-point-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

액세스 포인트 정책을 검색하려면

다음 get-access-point-policy 예제는 finance-ap 계정 123456789012에 이름이 지정된 액세스 포인트에서 액세스 포인트 정책을 검색합니다. 이 예제를 실행하기 전에 액세스 포인트 이름과 계정 번호를 사용 사례에 적합한 값으로 바꿉니다.

```
aws s3control get-access-point-policy \
    --account-id 123456789012 \
    --name finance-ap
```

출력:

```
{
```

```
    "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/*\"}]}"
}
```

자세한 내용은 Amazon Simple Storage Service 개발자 안내서의 Amazon S3 액세스 포인트를 사용한 데이터 액세스 관리를 참조하세요.

- 자세한 API 내용은 명령 참조GetAccessPointPolicy의 섹션을 참조하세요. AWS CLI

### get-access-point

다음 코드 예시에서는 get-access-point을 사용하는 방법을 보여 줍니다.

AWS CLI

액세스 포인트 구성 세부 정보를 검색하려면

다음 get-access-point 예제에서는 계정 123456789012finance-ap에 이름이 지정된 액세스 포인트의 구성 세부 정보를 검색합니다. 이 예제를 실행하기 전에 액세스 포인트 이름과 계정 번호를 사용 사례에 적합한 값으로 바꿉니다.

```
aws s3control get-access-point \
    --account-id 123456789012 \
    --name finance-ap
```

출력:

```
{
    "Name": "finance-ap",
    "Bucket": "business-records",
    "NetworkOrigin": "Internet",
    "PublicAccessBlockConfiguration": {
        "BlockPublicAcls": false,
        "IgnorePublicAcls": false,
        "BlockPublicPolicy": false,
        "RestrictPublicBuckets": false
    },
    "CreationDate": "2020-01-01T00:00:00Z"
}
```

자세한 내용은 Amazon Simple Storage Service 개발자 안내서의 Amazon S3 액세스 포인트를 사용한 데이터 액세스 관리를 참조하세요.

- 자세한 API 내용은 명령 참조GetAccessPoint의 섹션을 참조하세요. AWS CLI

### get-multi-region-access-point-routes

다음 코드 예시에서는 get-multi-region-access-point-routes을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 다중 리전 액세스 포인트 라우팅 구성을 쿼리하려면

다음 get-multi-region-access-point-routes 예제에서는 지정된 다중 리전 액세스 포인트에 대한 현재 라우팅 구성을 반환합니다.

```
aws s3control get-multi-region-access-point-routes \
    --region Region \
    --account-id 111122223333 \
    --mrap MultiRegionAccessPoint_ARN
```

출력:

```
{
    "Mrap": "arn:aws:s3::111122223333:accesspoint/0000000000000.mrap",
    "Routes": [
        {
            "Bucket": "DOC-EXAMPLE-BUCKET-1",
            "Region": "ap-southeast-2",
            "TrafficDialPercentage": 100
        },
        {
            "Bucket": "DOC-EXAMPLE-BUCKET-2",
            "Region": "us-west-1",
            "TrafficDialPercentage": 0
        }
    ]
}
```

- 자세한 API 내용은 명령 참조GetMultiRegionAccessPointRoutes의 섹션을 참조하세요. AWS CLI

### get-public-access-block

다음 코드 예시에서는 get-public-access-block을 사용하는 방법을 보여 줍니다.

AWS CLI

계정에 대한 퍼블릭 액세스 차단 설정을 나열하려면

다음 get-public-access-block 예제에서는 지정된 계정에 대한 퍼블릭 액세스 차단 설정을 표시합니다.

```
aws s3control get-public-access-block \
    --account-id 123456789012
```

출력:

```
{
   "PublicAccessBlockConfiguration": {
      "BlockPublicPolicy": true,
      "RestrictPublicBuckets": true,
      "IgnorePublicAcls": true,
      "BlockPublicAcls": true
   }
}
```

- 자세한 API 내용은 명령 참조GetPublicAccessBlock의 섹션을 참조하세요. AWS CLI

### list-access-points

다음 코드 예시에서는 list-access-points을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 계정의 모든 액세스 포인트 목록을 검색하려면

다음 list-access-points 예제에서는 계정 123456789012에서 소유한 버킷에 연결된 모든 액세스 포인트의 목록을 표시합니다.

```
aws s3control list-access-points \
    --account-id 123456789012
```

출력:

```
{
    "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"
        }
    ]
}
```

예제 2: 버킷의 모든 액세스 포인트 목록을 검색하려면

다음 `list-access-points` 예제에서는 계정 123456789012에서 `external-docs` 소유한 버킷에 연결된 모든 액세스 포인트 목록을 검색합니다.

```
aws s3control list-access-points \
    --account-id 123456789012 \
    --bucket external-docs
```

출력:

```
{
    "AccessPointList": [
        {
            "Name": "customer-ap",
            "NetworkOrigin": "Internet",
            "Bucket": "external-docs"
        },
        {
            "Name": "public-ap",
            "NetworkOrigin": "Internet",
            "Bucket": "external-docs"
        }
    ]
}
```

자세한 내용은 Amazon Simple Storage Service 개발자 안내서의 Amazon S3 액세스 포인트를 사용한 데이터 액세스 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ListAccessPoints의 섹션을 참조하세요. AWS CLI

### `list-jobs`

다음 코드 예시에서는 `list-jobs`을 사용하는 방법을 보여 줍니다.

AWS CLI

계정 Amazon S3 배치 작업 작업을 나열하려면

다음 `list-jobs` 예제에서는 지정된 계정에 대한 모든 최근 배치 작업 작업을 나열합니다.

```
aws s3control list-jobs \
    --account-id 123456789012
```

출력:

```
{
    "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
        },
    ]
}
```

- 자세한 API 내용은 명령 참조ListJobs의 섹션을 참조하세요. AWS CLI

### put-access-point-policy

다음 코드 예시에서는 put-access-point-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

액세스 포인트 정책을 설정하려면

다음 put-access-point-policy 예제에서는 계정 123456789012의 액세스 포인트finance-ap에 대해 지정된 액세스 포인트 정책을 배치합니다. 액세스 포인트에 finance-ap 이미 정책이 있는 경우 이 명령은 기존 정책을 이 명령에 지정된 정책으로 바꿉니다. 이 예제를 실행하기 전에 계정 번호, 액세스 포인트 이름 및 정책 문을 사용 사례에 적합한 값으로 바꿉니다.

```
aws s3control put-access-point-policy \
    --account-id 123456789012 \
    --name finance-ap \
```

```
    --policy file://ap-policy.json
```

ap-policy.json의 콘텐츠:

```
{
    "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/*"
        }
    ]
}
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Simple Storage Service 개발자 안내서의 Amazon S3 액세스 포인트를 사용한 데이터 액세스 관리를 참조하세요.

- 자세한 API 내용은 명령 참조PutAccessPointPolicy의 섹션을 참조하세요. AWS CLI

### put-public-access-block

다음 코드 예시에서는 put-public-access-block을 사용하는 방법을 보여 줍니다.

AWS CLI

계정의 퍼블릭 액세스 차단 설정을 편집하려면

다음 put-public-access-block 예제에서는 모든 퍼블릭 액세스 차단 설정을 지정된 계정에 true 대해 로 전환합니다.

```
aws s3control put-public-access-block \
    --account-id 123456789012 \
    --public-access-block-configuration '{"BlockPublicAcls": true,
 "IgnorePublicAcls": true, "BlockPublicPolicy": true, "RestrictPublicBuckets":
 true}'
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조PutPublicAccessBlock의 섹션을 참조하세요. AWS CLI

### submit-multi-region-access-point-routes

다음 코드 예시에서는 submit-multi-region-access-point-routes을 사용하는 방법을 보여 줍니다.

AWS CLI

다중 리전 액세스 포인트 라우팅 구성을 업데이트하려면

다음 submit-multi-region-access-point-routes 예제에서는 다중 ap-southeast-2 리전 액세스 포인트의 리전DOC-EXAMPLE-BUCKET-2에서 DOC-EXAMPLE-BUCKET-1 및 의 라우팅 상태를 업데이트합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조SubmitMultiRegionAccessPointRoutes의 섹션을 참조하세요. AWS CLI

### update-job-priority

다음 코드 예시에서는 update-job-priority을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon S3 배치 작업 작업의 작업 우선 순위를 업데이트하려면

다음 update-job-priority 예제에서는 지정된 작업을 새 우선 순위로 업데이트합니다.

```
aws s3control update-job-priority \
```

```
    --account-id 123456789012 \
    --job-id 8d9a18fe-c303-4d39-8ccc-860d372da386 \
    --priority 52
```

출력:

```
{
    "JobId": "8d9a18fe-c303-4d39-8ccc-860d372da386",
    "Priority": 52
}
```

- 자세한 API 내용은 명령 참조UpdateJobPriority의 섹션을 참조하세요. AWS CLI

### update-job-status

다음 코드 예시에서는 update-job-status을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon S3 배치 작업의 상태를 업데이트하려면

다음 update-job-status 예제에서는 승인을 기다리는 지정된 작업을 취소합니다.

```
aws s3control update-job-status \
    --account-id 123456789012 \
    --job-id 8d9a18fe-c303-4d39-8ccc-860d372da386 \
    --requested-job-status Cancelled
```

출력:

```
{
    "Status": "Cancelled",
    "JobId": "8d9a18fe-c303-4d39-8ccc-860d372da386"
}
```

다음 update-job-status 예제에서는 승인을 기다리는 지정된 를 확인하고 실행합니다.

```
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"
}
```

다음 update-job-status 예제에서는 실행 중인 지정된 작업을 취소합니다.

```
 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"
}
```

- 자세한 API 내용은 명령 참조UpdateJobStatus의 섹션을 참조하세요. AWS CLI

## 를 사용한 S3 Glacier 예제 AWS CLI

다음 코드 예제에서는 S3 Glacier와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### abort-multipart-upload

다음 코드 예시에서는 abort-multipart-upload을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 라는 이름의 저장소에 대한 진행 중인 멀티파트 업로드를 삭제합니다my-vault.

```
aws glacier abort-multipart-upload --account-id - --vault-name my-vault
 --upload-id 19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-
OssZtLqyFu7sY1_lR7vgFuJV6NtcV5zpsJ
```

이 명령은 출력을 생성하지 않습니다. Amazon Glacier에서는 작업을 수행할 때 계정 ID 인수가 필요하지만 하이픈을 사용하여 사용 중인 계정을 지정할 수 있습니다. 업로드 ID는 aws glacier initiate-multipart-upload 명령으로 반환되며 aws glacier list-multipart-uploads를 사용하여 가져올 수도 있습니다.

를 사용하여 Amazon Glacier에 멀티파트 업로드하는 방법에 대한 자세한 내용은 AWS CLI 사용 설명서의 Amazon Glacier 사용을 AWS CLI참조하세요.

- 자세한 API 내용은 명령 참조AbortMultipartUpload의 섹션을 참조하세요. AWS CLI

### abort-vault-lock

다음 코드 예시에서는 abort-vault-lock을 사용하는 방법을 보여 줍니다.

AWS CLI

진행 중인 볼트 잠금 프로세스를 중단하려면

다음 abort-vault-lock 예제에서는 지정된 저장소에서 저장소 잠금 정책을 삭제하고 저장소 잠금의 잠금 상태를 잠금 해제로 재설정합니다.

```
aws glacier abort-vault-lock \
    --account-id - \
    --vault-name MyVaultName
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Glacier API 개발자 안내서의 볼트 잠금 중단(DELETE 잠금 정책)을 참조하세요.

- 자세한 API 내용은 명령 참조AbortVaultLock의 섹션을 참조하세요. AWS CLI

### add-tags-to-vault

다음 코드 예시에서는 add-tags-to-vault을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 이름이 지정된 my-vault에 두 개의 태그를 추가합니다.

```
aws glacier add-tags-to-vault --account-id - --vault-name my-vault --
tags id=1234,date=july2015
```

Amazon Glacier에서는 작업을 수행할 때 계정 ID 인수가 필요하지만 하이픈을 사용하여 사용 중인 계정을 지정할 수 있습니다.

- 자세한 API 내용은 명령 참조AddTagsToVault의 섹션을 참조하세요. AWS CLI

### complete-multipart-upload

다음 코드 예시에서는 complete-multipart-upload을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 3MiB 아카이브에 대한 멀티파트 업로드를 완료합니다.

```
aws glacier complete-multipart-upload --archive-size 3145728 --
checksum 9628195fcdbcbbe76cdde456d4646fa7de5f219fb39823836d81f0cc0e18aa67
 --upload-id 19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-
OssZtLqyFu7sY1_lR7vgFuJV6NtcV5zpsJ --account-id - --vault-name my-vault
```

Amazon Glacier에서는 작업을 수행할 때 계정 ID 인수가 필요하지만 하이픈을 사용하여 사용 중인 계정을 지정할 수 있습니다.

업로드 ID는 aws glacier initiate-multipart-upload 명령으로 반환되며 aws glacier list-multipart-uploads를 사용하여 가져올 수도 있습니다. 체크섬 파라미터는 아카이브의 SHA-256 트리 해시를 16진수로 사용합니다.

트리 해시 계산 지침을 AWS CLI포함하여 를 사용하여 Amazon Glacier에 멀티파트 업로드하는 방법에 대한 자세한 내용은 AWS CLI 사용 설명서의 Amazon Glacier 사용을 참조하세요.

- 자세한 API 내용은 명령 참조CompleteMultipartUpload의 섹션을 참조하세요. AWS CLI

### complete-vault-lock

다음 코드 예시에서는 complete-vault-lock을 사용하는 방법을 보여 줍니다.

AWS CLI

진행 중인 볼트 잠금 프로세스를 완료하려면

다음 complete-vault-lock 예제에서는 지정된 저장소의 진행 중인 잠금 진행 상황을 완료하고 저장소 잠금의 잠금 상태를 로 설정합니다Locked. 를 실행할 때 lock-id 파라미터 값을 가져옵니다initiate-lock-process.

```
aws glacier complete-vault-lock \
    --account-id - \
    --vault-name MyVaultName \
    --lock-id 9QZgEXAMPLEPhvL6xEXAMPLE
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Glacier API 개발자 안내서의 전체 볼트 잠금(POST lockId)을 참조하세요.

- 자세한 API 내용은 명령 참조CompleteVaultLock의 섹션을 참조하세요. AWS CLI

### create-vault

다음 코드 예시에서는 create-vault을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 my-vault라는 새 볼트를 생성합니다.

```
aws glacier create-vault --vault-name my-vault --account-id -
```

Amazon Glacier에서는 작업을 수행할 때 계정 ID 인수가 필요하지만 하이픈을 사용하여 사용 중인 계정을 지정할 수 있습니다.

- 자세한 API 내용은 명령 참조CreateVault의 섹션을 참조하세요. AWS CLI

### delete-archive

다음 코드 예시에서는 delete-archive을 사용하는 방법을 보여 줍니다.

AWS CLI

볼트에서 아카이브를 삭제하는 방법

다음 delete-archive 예시에서는 example_vault에서 지정된 아카이브를 제거합니다.

```
aws glacier delete-archive \
    --account-id 111122223333  \
    --vault-name example_vault \
    --archive-id Sc0u9ZP8yaWkmh-XGlIvAVprtLhaLCGnNwNl5I5x9HqPIkX5mjc0DrId3Ln-
Gi_k2HzmlIDZUz117KSdVMdMXLuFWi9PJUitxWO73edQ43eTlMWkH0pd9zVSAuV_XXZBVhKhyGhJ7w
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeleteArchive의 섹션을 참조하세요. AWS CLI

### delete-vault-access-policy

다음 코드 예시에서는 delete-vault-access-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

볼트의 액세스 정책을 제거하려면

다음 delete-vault-access-policy 예제에서는 지정된 볼트에 대한 액세스 정책을 제거합니다.

```
aws glacier delete-vault-access-policy \
    --account-id 111122223333 \
    --vault-name example_vault
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeleteVaultAccessPolicy의 섹션을 참조하세요. AWS CLI

### delete-vault-notifications

다음 코드 예시에서는 delete-vault-notifications을 사용하는 방법을 보여 줍니다.

AWS CLI

저장소에 대한 SNS 알림을 제거하려면

다음 `delete-vault-notifications` 예제에서는 지정된 저장소에 대해 Amazon Simple Notification Service(Amazon SNS)에서 보낸 알림을 제거합니다.

```
aws glacier delete-vault-notifications \
    --account-id 111122223333 \
    --vault-name example_vault
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeleteVaultNotifications의 섹션을 참조하세요. AWS CLI

### `delete-vault`

다음 코드 예시에서는 `delete-vault`을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 `my-vault`라는 볼트를 삭제합니다.

```
aws glacier delete-vault --vault-name my-vault --account-id -
```

이 명령은 출력을 생성하지 않습니다. Amazon Glacier에서는 작업을 수행할 때 계정 ID 인수가 필요하지만 하이픈을 사용하여 사용 중인 계정을 지정할 수 있습니다.

- 자세한 API 내용은 명령 참조DeleteVault의 섹션을 참조하세요. AWS CLI

### `describe-job`

다음 코드 예시에서는 `describe-job`을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 `my-vault`라는 저장소의 인벤토리 검색 작업에 대한 정보를 검색합니다.

```
aws glacier describe-job --account-id - --vault-name my-
vault --job-id zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-
RO47Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_XqlNHS61ds04CnMW
```

출력:

```
{
    "InventoryRetrievalParameters": {
        "Format": "JSON"
    },
    "VaultARN": "arn:aws:glacier:us-west-2:0123456789012:vaults/my-vault",
    "Completed": false,
    "JobId": "zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-
RO47Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_XqlNHS61dsO4CnMW",
    "Action": "InventoryRetrieval",
    "CreationDate": "2015-07-17T20:23:41.616Z",
    "StatusCode": "InProgress"
}
```

작업 ID는 `aws glacier initiate-job` 및 `aws glacier list-jobs`의 출력에서 찾을 수 있습니다. Amazon Glacier에서는 작업을 수행할 때 계정 ID 인수가 필요하지만 하이픈을 사용하여 사용 중인 계정을 지정할 수 있습니다.

- 자세한 API 내용은 명령 참조DescribeJob의 섹션을 참조하세요. AWS CLI

### `describe-vault`

다음 코드 예시에서는 `describe-vault`을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 `my-vault`라는 볼트에 대한 데이터를 검색합니다.

```
aws glacier describe-vault --vault-name my-vault --account-id -
```

Amazon Glacier에서는 작업을 수행할 때 계정 ID 인수가 필요하지만 하이픈을 사용하여 사용 중인 계정을 지정할 수 있습니다.

- 자세한 API 내용은 명령 참조DescribeVault의 섹션을 참조하세요. AWS CLI

### `get-data-retrieval-policy`

다음 코드 예시에서는 `get-data-retrieval-policy`을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 사용 중인 계정에 대한 데이터 검색 정책을 가져옵니다.

```
aws glacier get-data-retrieval-policy --account-id -
```

출력:

```
{
    "Policy": {
        "Rules": [
            {
                "BytesPerHour": 10737418240,
                "Strategy": "BytesPerHour"
            }
        ]
    }
}
```

Amazon Glacier에서는 작업을 수행할 때 계정 ID 인수가 필요하지만 하이픈을 사용하여 사용 중인 계정을 지정할 수 있습니다.

- 자세한 API 내용은 명령 참조GetDataRetrievalPolicy의 섹션을 참조하세요. AWS CLI

### get-job-output

다음 코드 예시에서는 get-job-output을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 볼트 인벤토리 작업의 출력을 output.json라는 현재 디렉터리의 파일에 저장합니다.

```
aws glacier get-job-output --account-id - --vault-name my-
vault --job-id zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-
RO47Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_XqlNHS61ds04CnMW output.json
```

job-id는 aws glacier list-jobs의 출력에서 확인할 수 있습니다. 참고로 출력 파일 이름은 옵션 이름이 접두사로 붙지 않는 위치 인수입니다. Amazon Glacier에서는 작업을 수행할 때 계정 ID 인수가 필요하지만 하이픈을 사용하여 사용 중인 계정을 지정할 수 있습니다.

출력:

```
{
    "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":"kKB7ymWJVpPSwhGP6ycSOAekp9ZYe_--zM_mw6k76ZFGEIWQX-
ybtRDvc2VkPSDtfKmQrj0IRQLSGsNuDp-
AJVlu2ccmDSyDUmZwKbwbpAdGATGDiB3hHO0bjbGehXTcApVud_wyDw","ArchiveDescription":"multipart
 upload
 test","CreationDate":"2015-04-06T22:24:34Z","Size":3145728,"SHA256TreeHash":"9628195fcdbcbb
```

- 자세한 API 내용은 명령 참조GetJobOutput의 섹션을 참조하세요. AWS CLI

### get-vault-access-policy

다음 코드 예시에서는 get-vault-access-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

볼트의 액세스 정책을 검색하려면

다음 get-vault-access-policy 예제에서는 지정된 볼트에 대한 액세스 정책을 검색합니다.

```
aws glacier get-vault-access-policy \
    --account-id 111122223333 \
    --vault-name example_vault
```

출력:

```
{
    "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\"}]}"
    }
}
```

- 자세한 API 내용은 명령 참조GetVaultAccessPolicy의 섹션을 참조하세요. AWS CLI

### get-vault-lock

다음 코드 예시에서는 get-vault-lock을 사용하는 방법을 보여 줍니다.

AWS CLI

볼트 잠금의 세부 정보를 가져오려면

다음 get-vault-lock 예제에서는 지정된 볼트의 잠금에 대한 세부 정보를 검색했습니다.

```
aws glacier get-vault-lock \
    --account-id - \
    --vault-name MyVaultName
```

출력:

```
{
    "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-2:99999999999:vaults/MyVaultName\",\"Condition\":{\"NumericLessThanEquals\":
{\"glacier:ArchiveAgeinDays\":\"365\"}}}]}",
    "State": "Locked",
    "CreationDate": "2019-07-29T22:25:28.640Z"
}
```

자세한 내용은 Amazon Glacier API 개발자 안내서의 Get Vault Lock(GET lock-policy)을 참조하세요.

- 자세한 API 내용은 명령 참조GetVaultLock의 섹션을 참조하세요. AWS CLI

### get-vault-notifications

다음 코드 예시에서는 get-vault-notifications을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 my-vault라는 저장소의 알림 구성 설명을 가져옵니다.

```
aws glacier get-vault-notifications --account-id - --vault-name my-vault
```

출력:

```
{
    "vaultNotificationConfig": {
        "Events": [
            "InventoryRetrievalCompleted",
            "ArchiveRetrievalCompleted"
        ],
        "SNSTopic": "arn:aws:sns:us-west-2:0123456789012:my-vault"
    }
}
```

볼트에 대한 알림이 구성되지 않은 경우에는 오류가 반환됩니다. Amazon Glacier에서는 작업을 수행할 때 계정 ID 인수가 필요하지만 하이픈을 사용하여 사용 중인 계정을 지정할 수 있습니다.

- 자세한 API 내용은 명령 참조GetVaultNotifications의 섹션을 참조하세요. AWS CLI

### initiate-job

다음 코드 예시에서는 initiate-job을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 볼트 인벤토리를 가져오는 작업을 시작합니다my-vault.

```
aws glacier initiate-job --account-id - --vault-name my-vault --job-parameters
 '{"Type": "inventory-retrieval"}'
```

출력:

```
{
    "location": "/0123456789012/vaults/my-vault/jobs/
zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-
RO47Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_XqlNHS61dsO4CnMW",
    "jobId": "zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-
RO47Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_XqlNHS61dsO4CnMW"
```

```
}
```

Amazon Glacier에서는 작업을 수행할 때 계정 ID 인수가 필요하지만 하이픈을 사용하여 사용 중인 계정을 지정할 수 있습니다.

다음 명령은 볼트 에서 아카이브를 검색하는 작업을 시작합니다my-vault.

```
aws glacier initiate-job --account-id - --vault-name my-vault --job-
parameters file://job-archive-retrieval.json
```

job-archive-retrieval.json 는 작업 유형, 아카이브 ID 및 일부 선택적 파라미터를 지정하는 로컬 폴더의 JSON 파일입니다.

```
{
  "Type": "archive-retrieval",
  "ArchiveId": "kKB7ymWJVpPSwhGP6ycSOAekp9ZYe_--zM_mw6k76ZFGEIWQX-
ybtRDvc2VkPSDtfKmQrj0IRQLSGsNuDp-
AJVlu2ccmDSyDUmZwKbwbpAdGATGDiB3hHO0bjbGehXTcApVud_wyDw",
  "Description": "Retrieve archive on 2015-07-17",
  "SNSTopic": "arn:aws:sns:us-west-2:0123456789012:my-topic"
}
```

아카이브IDs는 aws glacier upload-archive 및 출력에서 사용할 수 있습니다aws glacier get-job-output.

출력:

```
{
    "location": "/011685312445/vaults/mwunderl/jobs/l7IL5-
EkXyEY9Ws95fClzIbk2O5uLYaFdAYOi-
azsX_Z8V6NH4yERHzars8wTKYQMX6nBDI9cMNHzyZJO59-8N9aHWav",
    "jobId": "l7IL5-EkXy2O5uLYaFdAYOiEY9Ws95fClzIbk-
azsX_Z8V6NH4yERHzars8wTKYQMX6nBDI9cMNHzyZJO59-8N9aHWav"
}
```

작업 파라미터 형식에 대한 자세한 내용은 Amazon Glacier API 참조의 작업 시작을 참조하세요.

- 자세한 API 내용은 명령 참조InitiateJob의 섹션을 참조하세요. AWS CLI

### initiate-multipart-upload

다음 코드 예시에서는 initiate-multipart-upload을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 파일당 1MiB(1,024 x 1,024바이트)의 부분 크기my-vault로 라는 볼트에 대한 멀티파트 업로드를 시작합니다.

```
aws glacier initiate-multipart-upload --account-id - --part-size 1048576 --vault-
name my-vault --archive-description "multipart upload test"
```

아카이브 설명 파라미터는 선택 사항입니다. Amazon Glacier에서는 작업을 수행할 때 계정 ID 인수가 필요하지만 하이픈을 사용하여 사용 중인 계정을 지정할 수 있습니다.

이 명령은 성공하면 업로드 ID를 출력합니다. 를 사용하여 아카이브의 각 부분을 업로드할 때 업로드 ID를 사용합니다aws glacier upload-multipart-part. 를 사용하여 Amazon Glacier에 멀티파트 업로드하는 방법에 대한 자세한 내용은 AWS CLI 사용 설명서의 Amazon Glacier 사용을 AWS CLI참조하세요.

- 자세한 API 내용은 명령 참조InitiateMultipartUpload의 섹션을 참조하세요. AWS CLI

### initiate-vault-lock

다음 코드 예시에서는 initiate-vault-lock을 사용하는 방법을 보여 줍니다.

AWS CLI

볼트 잠금 프로세스를 시작하려면

다음 initiate-vault-lock 예제에서는 지정된 저장소에 저장소 잠금 정책을 설치하고 저장소 잠금의 잠금 상태를 로 설정합니다InProgress. 저장소 잠금 상태를 로 설정하려면 24시간 complete-vault-lock 이내에 를 호출하여 프로세스를 완료해야 합니다Locked.

```
aws glacier initiate-vault-lock \
    --account-id - \
    --vault-name MyVaultName \
    --policy file://vault_lock_policy.json
```

vault_lock_policy.json의 콘텐츠:

```
{"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\"}}}]}"}
```

출력은 저장소 잠금 프로세스를 완료하는 데 사용할 수 있는 저장소 잠금 ID입니다.

```
{
    "lockId": "9QZgEXAMPLEPhvL6xEXAMPLE"
}
```

자세한 내용은 Amazon Glacier API 개발자 안내서의 볼트 잠금 시작(POST 잠금 정책)을 참조하세요.

- 자세한 API 내용은 명령 참조InitiateVaultLock의 섹션을 참조하세요. AWS CLI

### `list-jobs`

다음 코드 예시에서는 `list-jobs`을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 `my-vault`라는 볼트에 대해 진행 중인 작업과 최근에 완료된 작업을 나열합니다.

```
aws glacier list-jobs --account-id - --vault-name my-vault
```

출력:

```
{
    "JobList": [
        {
            "VaultARN": "arn:aws:glacier:us-west-2:0123456789012:vaults/my-vault",
            "RetrievalByteRange": "0-3145727",
            "SNSTopic": "arn:aws:sns:us-west-2:0123456789012:my-vault",
            "Completed": false,
            "SHA256TreeHash":
 "9628195fcdbcbbe76cdde932d4646fa7de5f219fb39823836d81f0cc0e18aa67",
            "JobId": "l7IL5-EkXyEY9Ws95fClzIbk2O5uLYaFdAYOi-
azsX_Z8V6NH4yERHzars8wTKYQMX6nBDI9cMNHzyZJO59-8N9aHWav",
            "ArchiveId": "kKB7ymWJVpPSwhGP6ycSOAekp9ZYe_--zM_mw6k76ZFGEIWQX-
ybtRDvc2VkPSDtfKmQrj0IRQLSGsNuDp-
AJVlu2ccmDSyDUmZwKbwbpAdGATGDiB3hHO0bjbGehXTcApVud_wyDw",
            "JobDescription": "Retrieve archive on 2015-07-17",
```

```
            "ArchiveSizeInBytes": 3145728,
            "Action": "ArchiveRetrieval",
            "ArchiveSHA256TreeHash":
 "9628195fcdbcbbe76cdde932d4646fa7de5f219fb39823836d81f0cc0e18aa67",
            "CreationDate": "2015-07-17T21:16:13.840Z",
            "StatusCode": "InProgress"
        },
        {
            "InventoryRetrievalParameters": {
                "Format": "JSON"
            },
            "VaultARN": "arn:aws:glacier:us-west-2:0123456789012:vaults/my-vault",
            "Completed": false,
            "JobId": "zbxcm3Z_3z5UkoroF7SuZKrxgGoDc3RloGduS7Eg-
RO47Yc6FxsdGBgf_Q2DK5Ejh18CnTS5XW4_XqlNHS61dsO4CnMW",
            "Action": "InventoryRetrieval",
            "CreationDate": "2015-07-17T20:23:41.616Z",
            "StatusCode": ""InProgress""
        }
    ]
}
```

Amazon Glacier에서는 작업을 수행할 때 계정 ID 인수가 필요하지만 하이픈을 사용하여 사용 중인 계정을 지정할 수 있습니다.

- 자세한 API 내용은 명령 참조ListJobs의 섹션을 참조하세요. AWS CLI

### `list-multipart-uploads`

다음 코드 예시에서는 `list-multipart-uploads`을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 라는 저장소에 대해 진행 중인 멀티파트 업로드를 모두 보여줍니다`my-vault`.

```
aws glacier list-multipart-uploads --account-id - --vault-name my-vault
```

Amazon Glacier에서는 작업을 수행할 때 계정 ID 인수가 필요하지만 하이픈을 사용하여 사용 중인 계정을 지정할 수 있습니다.

를 사용하여 Amazon Glacier에 멀티파트 업로드하는 방법에 대한 자세한 내용은 AWS CLI 사용 설명서의 Amazon Glacier 사용을 AWS CLI참조하세요.

- 자세한 API 내용은 명령 참조ListMultipartUploads의 섹션을 참조하세요. AWS CLI

### `list-parts`

다음 코드 예시에서는 `list-parts`을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 라는 저장소에 대한 멀티파트 업로드에 대해 업로드된 부분을 나열합니다`my-vault`.

```
aws glacier list-parts --account-id - --vault-name my-vault --upload-id "SYZi7qnL-
YGqGwAm8Kn3BLP2ElNCvnB-5961R09CSaPmPwkYGHOqeN_nX3-Vhnd2yF0KfB5FkmbnBU9GubbdrCs8ut-D"
```

출력:

```
{
    "MultipartUploadId": "SYZi7qnL-
YGqGwAm8Kn3BLP2ElNCvnB-5961R09CSaPmPwkYGHOqeN_nX3-Vhnd2yF0KfB5FkmbnBU9GubbdrCs8ut-
D",
    "Parts": [
        {
            "RangeInBytes": "0-1048575",
            "SHA256TreeHash":
 "e1f2a7cd6e047350f69b9f8cfa60fa606fe2f02802097a9a026360a7edc1f553"
        },
        {
            "RangeInBytes": "1048576-2097151",
            "SHA256TreeHash":
 "43cf3061fb95796aed99a11a6aa3cd8f839eed15e655ab0a597126210636aee6"
        }
    ],
    "VaultARN": "arn:aws:glacier:us-west-2:0123456789012:vaults/my-vault",
    "CreationDate": "2015-07-18T00:05:23.830Z",
    "PartSizeInBytes": 1048576
}
```

Amazon Glacier에서는 작업을 수행할 때 계정 ID 인수가 필요하지만 하이픈을 사용하여 사용 중인 계정을 지정할 수 있습니다.

를 사용하여 Amazon Glacier에 멀티파트 업로드하는 방법에 대한 자세한 내용은 AWS CLI 사용 설명서의 Amazon Glacier 사용을 AWS CLI참조하세요.

- 자세한 API 내용은 명령 참조ListParts의 섹션을 참조하세요. AWS CLI

### `list-provisioned-capacity`

다음 코드 예시에서는 `list-provisioned-capacity`을 사용하는 방법을 보여 줍니다.

AWS CLI

프로비저닝된 용량 단위를 검색하려면

다음 `list-provisioned-capacity` 예제에서는 지정된 계정에 대해 프로비저닝된 용량 단위에 대한 세부 정보를 검색합니다.

```
aws glacier list-provisioned-capacity \
    --account-id 111122223333
```

출력:

```
{
    "ProvisionedCapacityList": [
        {
            "CapacityId": "HpASAuvfRFiVDbOjMfEIcr8K",
            "ExpirationDate": "2020-03-18T19:59:24.000Z",
            "StartDate": "2020-02-18T19:59:24.912Z"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListProvisionedCapacity의 섹션을 참조하세요. AWS CLI

### `list-tags-for-vault`

다음 코드 예시에서는 `list-tags-for-vault`을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 `my-vault`라는 볼트에 적용된 태그를 나열합니다.

```
aws glacier list-tags-for-vault --account-id - --vault-name my-vault
```

출력:

```
{
    "Tags": {
        "date": "july2015",
        "id": "1234"
    }
}
```

Amazon Glacier에서는 작업을 수행할 때 계정 ID 인수가 필요하지만 하이픈을 사용하여 사용 중인 계정을 지정할 수 있습니다.

- 자세한 API 내용은 명령 참조ListTagsForVault의 섹션을 참조하세요. AWS CLI

### `list-vaults`

다음 코드 예시에서는 `list-vaults`을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 기본 계정 및 리전 내 볼트를 나열합니다.

```
aws glacier list-vaults --account-id -
```

출력:

```
{
    "VaultList": [
        {
            "SizeInBytes": 3178496,
            "VaultARN": "arn:aws:glacier:us-west-2:0123456789012:vaults/my-vault",
            "LastInventoryDate": "2015-04-07T00:26:19.028Z",
            "VaultName": "my-vault",
            "NumberOfArchives": 1,
            "CreationDate": "2015-04-06T21:23:45.708Z"
        }
    ]
}
```

Amazon Glacier에서는 작업을 수행할 때 계정 ID 인수가 필요하지만 하이픈을 사용하여 사용 중인 계정을 지정할 수 있습니다.

- 자세한 API 내용은 명령 참조ListVaults의 섹션을 참조하세요. AWS CLI

### purchase-provisioned-capacity

다음 코드 예시에서는 purchase-provisioned-capacity을 사용하는 방법을 보여 줍니다.

AWS CLI

프로비저닝된 용량 단위를 구매하려면

다음 purchase-provisioned-capacity 예제에서는 프로비저닝된 용량 단위를 구매합니다.

```
aws glacier purchase-provisioned-capacity \
    --account-id 111122223333
```

출력:

```
{
    "capacityId": "HpASAuvfRFiVDbOjMfEIcr8K"
}
```

- 자세한 API 내용은 명령 참조PurchaseProvisionedCapacity의 섹션을 참조하세요. AWS CLI

### remove-tags-from-vault

다음 코드 예시에서는 remove-tags-from-vault을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 이름이 인 볼트date에서 키가 있는 태그를 제거합니다my-vault.

```
aws glacier remove-tags-from-vault --account-id - --vault-name my-vault --tag-
keys date
```

Amazon Glacier에서는 작업을 수행할 때 계정 ID 인수가 필요하지만 하이픈을 사용하여 사용 중인 계정을 지정할 수 있습니다.

- 자세한 API 내용은 명령 참조RemoveTagsFromVault의 섹션을 참조하세요. AWS CLI

### set-data-retrieval-policy

다음 코드 예시에서는 set-data-retrieval-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 사용 중인 계정에 대한 데이터 검색 정책을 구성합니다.

```
aws glacier set-data-retrieval-policy --account-id - --policy file://data-retrieval-policy.json
```

data-retrieval-policy.json 는 데이터 검색 정책을 지정하는 현재 폴더의 JSON 파일입니다.

```
{
  "Rules":[
     {
         "Strategy":"BytesPerHour",
         "BytesPerHour":10737418240
      }
   ]
}
```

Amazon Glacier에서는 작업을 수행할 때 계정 ID 인수가 필요하지만 하이픈을 사용하여 사용 중인 계정을 지정할 수 있습니다.

다음 명령은 데이터 검색 정책을 인라인 FreeTier 사용 으로 설정합니다JSON.

```
aws glacier set-data-retrieval-policy --account-id - --policy '{"Rules":[{"Strategy":"FreeTier"}]}'
```

정책 형식에 대한 자세한 내용은 Amazon Glacier API 참조의 데이터 검색 정책 설정을 참조하세요.

- 자세한 API 내용은 명령 참조SetDataRetrievalPolicy의 섹션을 참조하세요. AWS CLI

### set-vault-access-policy

다음 코드 예시에서는 set-vault-access-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

볼트의 액세스 정책을 설정하려면

다음 set-vault-access-policy 예제에서는 권한 정책을 지정된 볼트에 연결합니다.

```
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\"}]}"}'
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조SetVaultAccessPolicy의 섹션을 참조하세요. AWS CLI

### set-vault-notifications

다음 코드 예시에서는 set-vault-notifications을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 이름이 인 볼트에 대한 SNS 알림을 구성합니다my-vault.

```
aws glacier set-vault-notifications --account-id - --vault-name my-vault --vault-
notification-config file://notificationconfig.json
```

notificationconfig.json 는 게시할 SNS 주제와 이벤트를 지정하는 현재 폴더의 JSON 파일입니다.

```
{
  "SNSTopic": "arn:aws:sns:us-west-2:0123456789012:my-vault",
  "Events": ["ArchiveRetrievalCompleted", "InventoryRetrievalCompleted"]
}
```

Amazon Glacier에서는 작업을 수행할 때 계정 ID 인수가 필요하지만 하이픈을 사용하여 사용 중인 계정을 지정할 수 있습니다.

- 자세한 API 내용은 명령 참조SetVaultNotifications의 섹션을 참조하세요. AWS CLI

### upload-archive

다음 코드 예시에서는 upload-archive을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 archive.zip이라는 현재 폴더의 아카이브를 my-vault라는 볼트에 업로드합니다.

```
aws glacier upload-archive --account-id - --vault-name my-vault --body archive.zip
```

출력:

```
{
    "archiveId": "kKB7ymWJVpPSwhGP6ycSOAekp9ZYe_--zM_mw6k76ZFGEIWQX-
ybtRDvc2VkPSDtfKmQrj0IRQLSGsNuDp-
AJVlu2ccmDSyDUmZwKbwbpAdGATGDiB3hHO0bjbGehXTcApVud_wyDw",
    "checksum": "969fb39823836d81f0cc028195fcdbcbbe76cdde932d4646fa7de5f21e18aa67",
    "location": "/0123456789012/vaults/my-vault/archives/
kKB7ymWJVpPSwhGP6ycSOAekp9ZYe_--zM_mw6k76ZFGEIWQX-ybtRDvc2VkPSDtfKmQrj0IRQLSGsNuDp-
AJVlu2ccmDSyDUmZwKbwbpAdGATGDiB3hHO0bjbGehXTcApVud_wyDw"
}
```

Amazon Glacier에서는 작업을 수행할 때 계정 ID 인수가 필요하지만 하이픈을 사용하여 사용 중인 계정을 지정할 수 있습니다.

업로드된 아카이브를 검색하려면 aws glacier initiate-job 명령을 사용하여 검색 작업을 시작하세요.

- 자세한 API 내용은 명령 참조UploadArchive의 섹션을 참조하세요. AWS CLI

### upload-multipart-part

다음 코드 예시에서는 upload-multipart-part을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 명령은 아카이브의 첫 번째 1MiB(1024 x 1024바이트) 부분을 업로드합니다.

```
aws glacier upload-multipart-part --body part1 --range 'bytes
 0-1048575/*' --account-id - --vault-name my-vault --upload-
id 19gaRezEXAMPLES6Ry5YYdqthHOC_kGRCT03L9yetr220UmPtBYKk-
OssZtLqyFu7sY1_lR7vgFuJV6NtcV5zpsJ
```

Amazon Glacier에서는 작업을 수행할 때 계정 ID 인수가 필요하지만 하이픈을 사용하여 사용 중인 계정을 지정할 수 있습니다.

본문 파라미터는 로컬 파일 시스템의 부분 파일 경로를 사용합니다. 범위 파라미터는 완료된 아카이브에서 파트가 차지하는 바이트를 나타내는 HTTP 콘텐츠 범위를 취합니다. 업로드 ID는 `aws glacier initiate-multipart-upload` 명령으로 반환되며 `aws glacier list-multipart-uploads`를 사용하여 가져올 수도 있습니다.

를 사용하여 Amazon Glacier에 멀티파트 업로드하는 방법에 대한 자세한 내용은 AWS CLI 사용 설명서의 Amazon Glacier 사용을 AWS CLI참조하세요.

- 자세한 API 내용은 명령 참조UploadMultipartPart의 섹션을 참조하세요. AWS CLI

## 를 사용한 Secrets Manager 예제 AWS CLI

다음 코드 예제에서는 Secrets Manager AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `batch-get-secret-value`

다음 코드 예시에서는 `batch-get-secret-value`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 이름으로 나열된 보안 암호 그룹의 보안 암호 값을 검색하려면

다음 `batch-get-secret-value` 예제에서는 세 개의 암호에 대한 암호 값 암호를 가져옵니다.

```
aws secretsmanager batch-get-secret-value \
    --secret-id-list MySecret1 MySecret2 MySecret3
```

출력:


```
{
    "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"
            ],
            "CreatedDate": "1523477145.729"
        },
        {
            "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret2-
a1b2c3",
            "Name": "MySecret2",
            "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbbb",
            "SecretString": "{\"username\":\"akua_mansa\",\"password\":\"EXAMPLE-
PASSWORD\"",
            "VersionStages": [
                "AWSCURRENT"
            ],
            "CreatedDate": "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"
            ],
            "CreatedDate": "1373477721.124"
        }
    ],
    "Errors": []
```

```
}
```

자세한 내용은 Secrets Manager 사용 설명서의 배치에서 보안 암호 그룹 검색을 참조하세요. AWS

예제 2: 필터로 선택한 보안 암호 그룹의 보안 암호 값을 검색하려면

다음 batch-get-secret-value 예제에서는 MySecret 이름에 가 있는 계정의 보안 암호 값 암호를 가져옵니다. 이름의 필터링은 대소문자를 구분합니다.

```
aws secretsmanager batch-get-secret-value \
    --filters Key="name",Values="MySecret"
```

출력:

```
{
    "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"
            ],
            "CreatedDate": "1523477145.729"
        },
        {
            "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret2-
a1b2c3",
            "Name": "MySecret2",
            "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbbb",
            "SecretString": "{\"username\":\"akua_mansa\",\"password\":\"EXAMPLE-
PASSWORD\"",
            "VersionStages": [
                "AWSCURRENT"
            ],
            "CreatedDate": "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"
            ],
            "CreatedDate": "1373477721.124"
        }
    ],
    "Errors": []
}
```

자세한 내용은 Secrets Manager 사용 설명서의 배치에서 보안 암호 그룹 검색을 참조하세요. AWS

- 자세한 API 내용은 명령 참조BatchGetSecretValue의 섹션을 참조하세요. AWS CLI

### cancel-rotate-secret

다음 코드 예시에서는 cancel-rotate-secret을 사용하는 방법을 보여 줍니다.

AWS CLI

보안 암호의 자동 교체를 끄려면

다음 cancel-rotate-secret 예제에서는 보안 암호의 자동 교체를 끕니다. 교체를 재개하려면 를 호출합니다rotate-secret.

```
aws secretsmanager cancel-rotate-secret \
    --secret-id MyTestSecret
```

출력:

```
{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-a1b2c3",
  "Name": "MyTestSecret"
}
```

자세한 내용은 Secrets Manager 사용 설명서의 암호 교체를 참조하세요.

- 자세한 API 내용은 명령 참조CancelRotateSecret의 섹션을 참조하세요. AWS CLI

### create-secret

다음 코드 예시에서는 create-secret을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: JSON 파일의 보안 인증 정보에서 보안 암호를 생성하려면

다음 create-secret 예시에서는 파일의 보안 인증 정보를 사용하여 보안 암호를 만듭니다. 자세한 내용은 AWS CLI 사용 설명서의 파일에서 파라미터 AWS CLI로드를 참조하세요.

```
aws secretsmanager create-secret \
    --name MyTestSecret \
    --secret-string file://mycreds.json
```

mycreds.json의 콘텐츠:

```
{
  "engine": "mysql",
  "username": "saanvis",
  "password": "EXAMPLE-PASSWORD",
  "host": "my-database-endpoint.us-west-2.rds.amazonaws.com",
  "dbname": "myDatabase",
  "port": "3306"
}
```

출력:

```
{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-a1b2c3",
  "Name": "MyTestSecret",
  "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

자세한 내용은 Secrets Manager 사용 설명서의 보안 암호 생성을 참조하세요.

예제 2: 보안 암호 생성

다음 create-secret 예시에서는 두 개의 키-값 쌍으로 보안 암호를 만듭니다. 명령 셸에 명령을 입력하면 명령 기록이 액세스되거나 유틸리티가 명령 파라미터에 액세스할 위험이 있습니다. 명령에 보안 암호 값이 포함된 경우 이는 문제가 됩니다. 자세한 내용은 Secrets Manager 사용 설명서의 명령줄 도구를 사용하여 보안 암호를 저장하는 위험 완화를 참조하세요.

```
aws secretsmanager create-secret \
    --name MyTestSecret \
    --description "My test secret created with the CLI." \
    --secret-string "{\"user\":\"diegor\",\"password\":\"EXAMPLE-PASSWORD\"}"
```

출력:

```
{
  "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-a1b2c3",
  "Name": "MyTestSecret",
  "VersionId": "EXAMPLE1-90ab-cdef-fedc-ba987EXAMPLE"
}
```

자세한 내용은 Secrets Manager 사용 설명서의 보안 암호 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateSecret의 섹션을 참조하세요. AWS CLI

### delete-resource-policy

다음 코드 예시에서는 delete-resource-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

보안 암호에 연결된 리소스 기반 정책을 삭제하려면

다음 delete-resource-policy 예시에서는 보안 암호에 연결된 리소스 기반 정책을 삭제합니다.

```
aws secretsmanager delete-resource-policy \
    --secret-id MyTestSecret
```

출력:

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
    "Name": "MyTestSecret"
}
```

자세한 내용은 Secrets Manager 사용 설명서의 인증 및 액세스 제어를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteResourcePolicy의 섹션을 참조하세요. AWS CLI

### delete-secret

다음 코드 예시에서는 delete-secret을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: 보안 암호를 삭제하는 방법

다음 delete-secret 예시에서는 보안 암호를 삭제합니다. DeletionDate 응답 필드의 날짜 및 시간까지 restore-secret으로 보안 암호를 복구할 수 있습니다. 다른 리전에 복제된 보안 암호를 삭제하려면 먼저 remove-regions-from-replication(으)로 해당 복제본을 삭제한 다음 delete-secret을(를) 호출합니다.

```
aws secretsmanager delete-secret \
    --secret-id MyTestSecret \
    --recovery-window-in-days 7
```

출력:

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
    "Name": "MyTestSecret",
    "DeletionDate": 1524085349.095
}
```

자세한 내용은 Secrets Manager 사용 설명서의 보안 암호 삭제를 참조하세요.

예 2: 보안 암호를 즉시 삭제하는 방법

다음 delete-secret 예시는 복구 기간 없이 즉시 보안 암호를 삭제합니다. 이러한 보안 암호는 복구할 수 없습니다.

```
aws secretsmanager delete-secret \
    --secret-id MyTestSecret \
    --force-delete-without-recovery
```

출력:

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
```

```
    "Name": "MyTestSecret",
    "DeletionDate": 1508750180.309
}
```

자세한 내용은 Secrets Manager 사용 설명서의 보안 암호 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteSecret의 섹션을 참조하세요. AWS CLI

### `describe-secret`

다음 코드 예시에서는 `describe-secret`을 사용하는 방법을 보여 줍니다.

AWS CLI

보안 암호의 세부 정보를 검색하는 방법

다음 `describe-secret` 예시에서는 보안 암호에 대한 세부 정보를 보여줍니다.

```
aws secretsmanager describe-secret \
    --secret-id MyTestSecret
```

출력:

```
{
    "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"
        ]
    },
    "CreatedDate": 1521534252.66,
    "PrimaryRegion": "us-west-2",
    "ReplicationStatus": [
        {
            "Region": "eu-west-3",
            "KmsKeyId": "alias/aws/secretsmanager",
            "Status": "InSync",
            "StatusMessage": "Replication succeeded"
        }
    ]
}
```

자세한 내용은 Secrets Manager 사용 설명서의 보안 암호를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeSecret의 섹션을 참조하세요. AWS CLI

### `get-random-password`

다음 코드 예시에서는 `get-random-password`을 사용하는 방법을 보여 줍니다.

AWS CLI

무작위 암호를 생성하려면

다음 get-random-password 예제에서는 대문자, 소문자, 숫자 및 구두점을 하나 이상 포함하는 20자 길이의 무작위 암호를 생성합니다.

```
aws secretsmanager get-random-password \
    --require-each-included-type \
    --password-length 20
```

출력:

```
{
    "RandomPassword": "EXAMPLE-PASSWORD"
}
```

자세한 내용은 Secrets Manager 사용 설명서의 보안 암호 생성 및 관리를 참조하세요.

- 자세한 API 내용은 명령 참조GetRandomPassword의 섹션을 참조하세요. AWS CLI

### get-resource-policy

다음 코드 예시에서는 get-resource-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

보안 암호에 연결된 리소스 기반 정책을 검색하려면

다음 get-resource-policy 예시에서는 보안 암호에 연결된 리소스 기반 정책을 검색합니다.

```
aws secretsmanager get-resource-policy \
    --secret-id MyTestSecret
```

출력:

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
    "Name": "MyTestSecret",
    "ResourcePolicy": "{\n\"Version\":\"2012-10-17\",\n\"Statement\":[{\n\"Effect\":
\"Allow\",\n
    \"Principal\":{\n\"AWS\":\"arn:aws:iam::123456789012:root\"\n},\n\"Action\":
    \"secretsmanager:GetSecretValue\",\n\"Resource\":\"*\"\n}]\n}"
```

```
}
```

자세한 내용은 Secrets Manager 사용 설명서의 인증 및 액세스 제어를 참조하세요.

- 자세한 API 내용은 명령 참조GetResourcePolicy의 섹션을 참조하세요. AWS CLI

### get-secret-value

다음 코드 예시에서는 get-secret-value을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: 보안 암호의 암호화된 보안 암호 값을 검색하는 방법

다음 get-secret-value 예에서는 현재 보안 암호 값을 가져옵니다.

```
aws secretsmanager get-secret-value \
    --secret-id MyTestSecret
```

출력:

```
{
    "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"
    ],
    "CreatedDate": 1523477145.713
}
```

자세한 내용은 Secrets Manager 사용 설명서의 보안 암호 검색을 참조하세요.

예 2: 이전 보안 암호 값 검색

다음 get-secret-value 예시에서는 이전 보안 암호 값을 가져옵니다.

```
aws secretsmanager get-secret-value \
    --secret-id MyTestSecret
```

```
    --version-stage AWSPREVIOUS
```

출력:

```
{
    "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"
    ],
    "CreatedDate": 1523477145.713
}
```

자세한 내용은 Secrets Manager 사용 설명서의 보안 암호 검색을 참조하세요.

- 자세한 API 내용은 명령 참조GetSecretValue의 섹션을 참조하세요. AWS CLI

### list-secret-version-ids

다음 코드 예시에서는 `list-secret-version-ids`을 사용하는 방법을 보여 줍니다.

AWS CLI

보안 암호와 연결된 모든 보안 암호 버전을 나열하려면

다음 `list-secret-version-ids` 예제에서는 보안 암호의 모든 버전 목록을 가져옵니다.

```
aws secretsmanager list-secret-version-ids \
    --secret-id MyTestSecret
```

출력:

```
{
  "Versions": [
    {
        "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "VersionStages": [
```

```
            "AWSPREVIOUS"
        ],
        "LastAccessedDate": 1523477145.713,
        "CreatedDate": 1523477145.713
    },
    {
        "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
        "VersionStages": [
            "AWSCURRENT"
        ],
        "LastAccessedDate": 1523477145.713,
        "CreatedDate": 1523486221.391
    },
    {
        "CreatedDate": 1.51197446236E9,
        "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333;"
    }
    ],
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
    "Name": "MyTestSecret"
}
```

자세한 내용은 Secrets Manager 사용 설명서의 버전을 참조하세요.

- 자세한 API 내용은 명령 참조ListSecretVersionIds의 섹션을 참조하세요. AWS CLI

### `list-secrets`

다음 코드 예시에서는 `list-secrets`을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: 계정의 보안 암호를 나열하는 방법

다음 `list-secrets` 예시에서는 계정에 있는 보안 암호 목록을 가져옵니다.

```
aws secretsmanager list-secrets
```

출력:

```
{
```

```
    "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"
                ]
            }
        }
    ]
}
```

자세한 내용은 Secrets Manager 사용 설명서의 보안 암호 찾기를 참조하세요.

예 1: 계정의 보안 암호 목록을 필터링하는 방법

다음 `list-secrets` 예시에서는 계정에서 이름에 `Test`가 있는 보안 암호 목록을 가져옵니다. 이름의 필터링은 대소문자를 구분합니다.

```
aws secretsmanager list-secrets \
    --filter Key="name",Values="Test"
```

출력:

```
{
    "SecretList": [
        {
            "ARN": "arn:aws:secretsmanager:us-
west-2:123456789012:secret:MyTestSecret-a1b2c3",
```

```
            "Name": "MyTestSecret",
            "LastChangedDate": 1523477145.729,
            "SecretVersionsToStages": {
                "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111": [
                    "AWSCURRENT"
                ]
            }
        }
    ]
}
```

자세한 내용은 Secrets Manager 사용 설명서의 보안 암호 찾기를 참조하세요.

예 3: 다른 서비스에서 관리하는 계정의 보안 암호를 나열하는 방법

다음 `list-secrets` 예제에서는 Amazon 에서 관리하는 계정의 보안 암호를 반환합니다RDS.

```
aws secretsmanager list-secrets \
    --filter Key="owning-service",Values="rds"
```

출력:

```
{
    "SecretList": [
        {
            "Name": "rds!cluster-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "Tags": [
                {
                    "Value": "arn:aws:rds:us-
west-2:123456789012:cluster:database-1",
                    "Key": "aws:rds:primaryDBClusterArn"
                },
                {
                    "Value": "rds",
                    "Key": "aws:secretsmanager:owningService"
                }
            ],
            "RotationRules": {
                "AutomaticallyAfterDays": 1
            },
            "LastChangedDate": 1673477781.275,
            "LastRotatedDate": 1673477781.26,
```

```
            "SecretVersionsToStages": {
                "a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa": [
                    "AWSPREVIOUS"
                ],
                "a1b2c3d4-5678-90ab-cdef-EXAMPLEbbbbb": [
                    "AWSCURRENT",
                    "AWSPENDING"
                ]
            },
            "OwningService": "rds",
            "RotationEnabled": true,
            "CreatedDate": 1673467300.7,
            "LastAccessedDate": 1673395200.0,
            "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:rds!
cluster-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111-a1b2c3",
            "Description": "Secret associated with primary RDS DB cluster:
 arn:aws:rds:us-west-2:123456789012:cluster:database-1"
        }
    ]
}
```

자세한 내용은 Secrets Manager 사용 설명서의 다른 서비스에서 관리하는 보안 암호를 참조하세요.

- 자세한 API 내용은 명령 참조ListSecrets의 섹션을 참조하세요. AWS CLI

### put-resource-policy

다음 코드 예시에서는 put-resource-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

보안 암호에 리소스 기반 정책을 추가하려면

다음 put-resource-policy 예시에서는 보안 암호에 사용 권한 정책을 추가하여 해당 정책이 암호에 대한 광범위한 액세스를 제공하지 않는지 먼저 확인합니다. 파일에서 해당 정책을 읽습니다. 자세한 내용은 AWS CLI 사용 설명서의 파일에서 파라미터 AWS CLI로드를 참조하세요.

```
aws secretsmanager put-resource-policy \
    --secret-id MyTestSecret \
    --resource-policy file://mypolicy.json \
    --block-public-policy
```

mypolicy.json의 콘텐츠:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:role/MyRole"
            },
            "Action": "secretsmanager:GetSecretValue",
            "Resource": "*"
        }
    ]
}
```

출력:

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
    "Name": "MyTestSecret"
}
```

자세한 내용은 Secrets Manager 사용 설명서의 보안 암호에 권한 정책 연결을 참조하세요.

- 자세한 API 내용은 명령 참조PutResourcePolicy의 섹션을 참조하세요. AWS CLI

### put-secret-value

다음 코드 예시에서는 put-secret-value을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: 보안 암호에 새 암호 값을 저장하는 방법

다음 put-secret-value 예시에서는 두 개의 키-값 페어로 새 버전의 보안 암호를 만듭니다.

```
aws secretsmanager put-secret-value \
    --secret-id MyTestSecret \
    --secret-string "{\"user\":\"diegor\",\"password\":\"EXAMPLE-PASSWORD\"}"
```

출력:

```
{
    "ARN": "arn:aws:secretsmanager:us-
west-2:123456789012:secret:MyTestSecret-1a2b3c",
    "Name": "MyTestSecret",
    "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "VersionStages": [
        "AWSCURRENT"
    ]
}
```

자세한 내용은 Secrets Manager 사용 설명서의 보안 암호 수정을 참조하세요.

예제 2: 자격 증명의 새 보안 암호 값을 JSON 파일에 저장하려면

다음 put-secret-value 예에서는 파일로 된 보안 인증 정보로 새 버전의 보안 암호를 만듭니다. 자세한 내용은 AWS CLI 사용 설명서의 파일에서 파라미터 AWS CLI로드를 참조하세요.

```
aws secretsmanager put-secret-value \
    --secret-id MyTestSecret \
    --secret-string file://mycreds.json
```

mycreds.json의 콘텐츠:

```
{
  "engine": "mysql",
  "username": "saanvis",
  "password": "EXAMPLE-PASSWORD",
  "host": "my-database-endpoint.us-west-2.rds.amazonaws.com",
  "dbname": "myDatabase",
  "port": "3306"
}
```

출력:

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
    "Name": "MyTestSecret",
    "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
```

```
    "VersionStages": [
        "AWSCURRENT"
    ]
}
```

자세한 내용은 Secrets Manager 사용 설명서의 보안 암호 수정을 참조하세요.

- 자세한 API 내용은 명령 참조PutSecretValue의 섹션을 참조하세요. AWS CLI

### `remove-regions-from-replication`

다음 코드 예시에서는 `remove-regions-from-replication`을 사용하는 방법을 보여 줍니다.

AWS CLI

복제본 보안 암호를 삭제하려면

다음 `remove-regions-from-replication` 예시에서는 eu-west-3의 복제 보안 암호를 삭제합니다. 다른 리전에 복제된 기본 보안 암호를 삭제하려면 먼저 복제본을 삭제한 다음 `delete-secret`을(를) 호출합니다.

```
aws secretsmanager remove-regions-from-replication \
    --secret-id MyTestSecret \
    --remove-replica-regions eu-west-3
```

출력:

```
{
    "ARN": "arn:aws:secretsmanager:us-
west-2:123456789012:secret:MyTestSecret-1a2b3c",
    "ReplicationStatus": []
}
```

자세한 내용은 Secrets Manager 사용 설명서의 복제본 보안 암호 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조RemoveRegionsFromReplication의 섹션을 참조하세요. AWS CLI

### `replicate-secret-to-regions`

다음 코드 예시에서는 `replicate-secret-to-regions`을 사용하는 방법을 보여 줍니다.

AWS CLI

보안 암호를 다른 리전에 복제하려면

다음 `replicate-secret-to-regions` 예시에서는 eu-west-3으로 보안 암호를 복제합니다. 복제본은 AWS 관리형 키 로 암호화됩니다`aws/secretsmanager`.

```
aws secretsmanager replicate-secret-to-regions \
    --secret-id MyTestSecret \
    --add-replica-regions Region=eu-west-3
```

출력:

```
{
    "ARN": "arn:aws:secretsmanager:us-
west-2:123456789012:secret:MyTestSecret-1a2b3c",
    "ReplicationStatus": [
        {
            "Region": "eu-west-3",
            "KmsKeyId": "alias/aws/secretsmanager",
            "Status": "InProgress"
        }
    ]
}
```

자세한 내용은 Secrets Manager 사용 설명서의 다른 리전에 보안 암호 복제를 참조하세요.

- 자세한 API 내용은 명령 참조ReplicateSecretToRegions의 섹션을 참조하세요. AWS CLI

### `restore-secret`

다음 코드 예시에서는 `restore-secret`을 사용하는 방법을 보여 줍니다.

AWS CLI

이전에 삭제한 보안 암호를 복원하려면

다음 `restore-secret` 예시에서는 이전에 삭제가 예정된 보안 암호를 복원합니다.

```
aws secretsmanager restore-secret \
    --secret-id MyTestSecret
```

출력:

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
    "Name": "MyTestSecret"
}
```

자세한 내용은 Secrets Manager 사용 설명서의 보안 암호 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조RestoreSecret의 섹션을 참조하세요. AWS CLI

### `rotate-secret`

다음 코드 예시에서는 `rotate-secret`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 보안 암호의 자동 교체를 구성하고 시작하려면

다음 `rotate-secret` 예제에서는 보안 암호의 자동 교체를 구성하고 시작합니다. Secrets Manager는 보안 암호를 즉시 한 번 교체한 다음 2시간 간격으로 8시간마다 교체합니다. 출력은 교체를 통해 생성된 새 보안 암호 버전의 `VersionId` 를 보여줍니다.

```
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\"}"
```

출력:

```
{
    "ARN": "aws:arn:secretsmanager:us-
west-2:123456789012:secret:MyTestDatabaseSecret-a1b2c3",
    "Name": "MyTestDatabaseSecret",
    "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

자세한 내용은 Secrets Manager 사용 설명서의 암호 교체를 참조하세요.

예제 2: 교체 간격에서 자동 교체를 구성하고 시작하려면

다음 `rotate-secret` 예제에서는 보안 암호의 자동 교체를 구성하고 시작합니다. Secrets Manager는 보안 암호를 즉시 한 번 교체한 다음 10일마다 교체합니다. 출력은 교체를 통해 생성된 새 보안 암호 버전의 `VersionId` 를 보여줍니다.

```
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)\"}"
```

출력:

```
{
    "ARN": "aws:arn:secretsmanager:us-
west-2:123456789012:secret:MyTestDatabaseSecret-a1b2c3",
    "Name": "MyTestDatabaseSecret",
    "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

자세한 내용은 Secrets Manager 사용 설명서의 암호 교체를 참조하세요.

예제 3: 암호를 즉시 교체하려면

다음 `rotate-secret` 예에서는 즉시 교체를 시작합니다. 출력은 교체를 통해 생성된 새 보안 암호 버전의 `VersionId` 를 보여줍니다. 보안 암호에 교체가 미리 구성되어 있어야 합니다.

```
aws secretsmanager rotate-secret \
    --secret-id MyTestDatabaseSecret
```

출력:

```
{
    "ARN": "aws:arn:secretsmanager:us-
west-2:123456789012:secret:MyTestDatabaseSecret-a1b2c3",
    "Name": "MyTestDatabaseSecret",
    "VersionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

자세한 내용은 Secrets Manager 사용 설명서의 암호 교체를 참조하세요.

- 자세한 API 내용은 명령 참조RotateSecret의 섹션을 참조하세요. AWS CLI

### stop-replication-to-replica

다음 코드 예시에서는 stop-replication-to-replica을 사용하는 방법을 보여 줍니다.

AWS CLI

복제본 암호를 기본 암호로 승격하려면

다음 stop-replication-to-replica 예시에서는 복제 암호와 기본 암호 간의 링크를 제거합니다. 복제 보안 암호는 복제본 리전의 기본 보안 암호로 승격됩니다. 복제 리전 내에서 stop-replication-to-replica을(를) 호출해야 합니다.

```
aws secretsmanager stop-replication-to-replica \
    --secret-id MyTestSecret
```

출력:

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3"
}
```

자세한 내용은 Secrets Manager 사용 설명서의 복제본 보안 암호 승격을 참조하세요.

- 자세한 API 내용은 명령 참조StopReplicationToReplica의 섹션을 참조하세요. AWS CLI

### tag-resource

다음 코드 예시에서는 tag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 보안 암호에 태그를 추가하려면

다음 예시에서는 간편 구문으로 태그를 연결하는 방법을 보여줍니다.

```
aws secretsmanager tag-resource \
    --secret-id MyTestSecret \
```

```
    --tags Key=FirstTag,Value=FirstValue
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Secrets Manager 사용 설명서의 보안 암호 태그 지정을 참조하세요.

예제 2: 보안 암호에 여러 태그를 추가하려면

다음 `tag-resource` 예시에서는 두 개의 키-값 태그를 보안 암호에 연결합니다.

```
aws secretsmanager tag-resource \
    --secret-id MyTestSecret \
    --tags '[{"Key": "FirstTag", "Value": "FirstValue"}, {"Key": "SecondTag",
 "Value": "SecondValue"}]'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Secrets Manager 사용 설명서의 보안 암호 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### `untag-resource`

다음 코드 예시에서는 `untag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

보안 암호에서 태그를 제거하려면

다음 `untag-resource` 예시에서는 보안 암호에서 두 개의 태그를 제거합니다. 각 태그의 키와 값이 모두 제거됩니다.

```
aws secretsmanager untag-resource \
    --secret-id MyTestSecret \
    --tag-keys '[ "FirstTag", "SecondTag"]'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Secrets Manager 사용 설명서의 보안 암호 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-secret-version-stage

다음 코드 예시에서는 update-secret-version-stage을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 보안 암호를 이전 버전으로 되돌리려면

다음 update-secret-version-stage 예제에서는 스테이징 레이블을 이전 버전의 보안 암호로 이동 AWS CURRENT하고, 이 보안 암호는 이전 버전으로 되돌립니다. 이전 버전의 ID를 찾으려면 를 사용합니다list-secret-version-ids. 이 예제에서 레이블이 AWS CURRENT 있는 버전은 a1b2c3d4-5678-90ab-cdef-EXAMPLE11111이고 레이블이 있는 AWS PREVIOUS 버전은 a1b2c3d4-5678-90ab-cdef-입니다EXAMPLE22222. 이 예제에서는 레이블을 AWS CURRENT 버전 11111에서 22222로 이동합니다. 레이블이 버전에서 제거되므로 AWS CURRENT 는 update-secret-version-stage AWS PREVIOUS 자동으로 레이블을 해당 버전(11111)으로 이동합니다. 그 효과는 AWS CURRENT 및 AWS PREVIOUS 버전이 교체된다는 것입니다.

```
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
```

출력:

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
    "Name": "MyTestSecret"
}
```

자세한 내용은 Secrets Manager 사용 설명서의 버전을 참조하세요.

예제 2: 보안 암호 버전에 연결된 스테이징 레이블을 추가하려면

다음 update-secret-version-stage 예제에서는 보안 암호 버전에 스테이징 레이블을 추가합니다. 영향을 받는 버전의 VersionStages 응답 필드를 실행list-secret-version-ids하고 확인하여 결과를 검토할 수 있습니다.

```
aws secretsmanager update-secret-version-stage \
```

```
    --secret-id MyTestSecret \
    --version-stage STAGINGLABEL1 \
    --move-to-version-id EXAMPLE1-90ab-cdef-fedc-ba987EXAMPLE
```

출력:

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
    "Name": "MyTestSecret"
}
```

자세한 내용은 Secrets Manager 사용 설명서의 버전을 참조하세요.

예제 3: 보안 암호 버전에 연결된 스테이징 레이블 삭제

다음 update-secret-version-stage 예제에서는 보안 암호 버전에 연결된 스테이징 레이블을 삭제합니다. 영향을 받는 버전의 VersionStages 응답 필드를 실행list-secret-version-ids하고 확인하여 결과를 검토할 수 있습니다.

```
aws secretsmanager update-secret-version-stage \
    --secret-id MyTestSecret \
    --version-stage STAGINGLABEL1 \
    --remove-from-version-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

출력:

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
    "Name": "MyTestSecret"
}
```

자세한 내용은 Secrets Manager 사용 설명서의 버전을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateSecretVersionStage의 섹션을 참조하세요. AWS CLI

### update-secret

다음 코드 예시에서는 update-secret을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: 보안 암호의 설명을 업데이트하는 방법

다음 update-secret 예에서는 보안 암호에 대한 설명을 업데이트합니다.

```
aws secretsmanager update-secret \
    --secret-id MyTestSecret \
    --description "This is a new description for the secret."
```

출력:

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
    "Name": "MyTestSecret"
}
```

자세한 내용은 Secrets Manager 사용 설명서의 보안 암호 수정을 참조하세요.

예 2: 보안 암호와 연결된 암호화 키를 업데이트하는 방법

다음 update-secret 예제에서는 보안 암호 값을 암호화하는 데 사용되는 KMS 키를 업데이트합니다. KMS 키는 보안 암호와 동일한 리전에 있어야 합니다.

```
aws secretsmanager update-secret \
    --secret-id MyTestSecret \
    --kms-key-id arn:aws:kms:us-west-2:123456789012:key/EXAMPLE1-90ab-cdef-fedc-
ba987EXAMPLE
```

출력:

```
{
    "ARN": "arn:aws:secretsmanager:us-west-2:123456789012:secret:MyTestSecret-
a1b2c3",
    "Name": "MyTestSecret"
}
```

자세한 내용은 Secrets Manager 사용 설명서의 보안 암호 수정을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateSecret의 섹션을 참조하세요. AWS CLI

### validate-resource-policy

다음 코드 예시에서는 validate-resource-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 정책을 검증하려면

다음 validate-resource-policy 예제에서는 리소스 정책이 보안 암호에 대한 광범위한 액세스 권한을 부여하지 않는지 확인합니다. 정책은 디스크의 파일에서 읽습니다. 자세한 내용은 AWS CLI 사용 설명서의 파일에서 파라미터 AWS CLI로드를 참조하세요.

```
aws secretsmanager validate-resource-policy \
    --resource-policy file://mypolicy.json
```

mypolicy.json의 콘텐츠:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:role/MyRole"
            },
            "Action": "secretsmanager:GetSecretValue",
            "Resource": "*"
        }
    ]
}
```

출력:

```
{
    "PolicyValidationPassed": true,
    "ValidationErrors": []
}
```

자세한 내용은 Secrets Manager 사용 설명서의 Secrets Manager에 대한 권한 참조를 참조하세요.

- 자세한 API 내용은 명령 참조ValidateResourcePolicy의 섹션을 참조하세요. AWS CLI

# 를 사용한 Security Hub 예제 AWS CLI

다음 코드 예제에서는 Security Hub AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `accept-administrator-invitation`

다음 코드 예시에서는 `accept-administrator-invitation`을 사용하는 방법을 보여 줍니다.

AWS CLI

관리자 계정에서 초대를 수락하려면

다음 `accept-administrator-invitation` 예제에서는 지정된 관리자 계정에서 지정된 초대를 수락합니다.

```
aws securityhub accept-invitation \
    --administrator-id 123456789012 \
    --invitation-id 7ab938c5d52d7904ad09f9e7c20cc4eb
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Security Hub 사용 설명서의 관리자 및 멤버 계정 관리를 참조하세요.

- 자세한 API 내용은 명령 참조AcceptAdministratorInvitation의 섹션을 참조하세요. AWS CLI

### `accept-invitation`

다음 코드 예시에서는 `accept-invitation`을 사용하는 방법을 보여 줍니다.

AWS CLI

관리자 계정에서 초대를 수락하려면

다음 accept-invitation 예제에서는 지정된 관리자 계정에서 지정된 초대를 수락합니다.

```
aws securityhub accept-invitation \
    --master-id 123456789012 \
    --invitation-id 7ab938c5d52d7904ad09f9e7c20cc4eb
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Security Hub 사용 설명서의 관리자 및 멤버 계정 관리를 참조하세요.

- 자세한 API 내용은 명령 참조AcceptInvitation의 섹션을 참조하세요. AWS CLI

## batch-delete-automation-rules

다음 코드 예시에서는 batch-delete-automation-rules을 사용하는 방법을 보여 줍니다.

AWS CLI

자동화 규칙을 삭제하려면

다음 batch-delete-automation-rules 예제에서는 지정된 자동화 규칙을 삭제합니다. 단일 명령으로 하나 이상의 규칙을 삭제할 수 있습니다. Security Hub 관리자 계정만 이 명령을 실행할 수 있습니다.

```
aws securityhub batch-delete-automation-rules \
    --automation-rules-arns '["arn:aws:securityhub:us-
east-1:123456789012:automation-rule/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"]'
```

출력:

```
{
    "ProcessedAutomationRules": [
        "arn:aws:securityhub:us-east-1:123456789012:automation-rule/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    ],
    "UnprocessedAutomationRules": []
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 자동화 규칙 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조BatchDeleteAutomationRules의 섹션을 참조하세요. AWS CLI

### batch-disable-standards

다음 코드 예시에서는 batch-disable-standards을 사용하는 방법을 보여 줍니다.

AWS CLI

표준을 비활성화하려면

다음 batch-disable-standards 예제에서는 지정된 구독 과 연결된 표준을 비활성화합니다 ARN.

```
aws securityhub batch-disable-standards \
    --standards-subscription-arns "arn:aws:securityhub:us-
west-1:123456789012:subscription/pci-dss/v/3.2.1"
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 보안 표준 비활성화 또는 활성화를 참조하세요.

- 자세한 API 내용은 명령 참조BatchDisableStandards의 섹션을 참조하세요. AWS CLI

### batch-enable-standards

다음 코드 예시에서는 batch-enable-standards을 사용하는 방법을 보여 줍니다.

AWS CLI

표준을 활성화하려면

다음 batch-enable-standards 예제에서는 요청 계정에 대한 PCI DSS 표준을 활성화합니다.

```
aws securityhub batch-enable-standards \
    --standards-subscription-requests '{"StandardsArn":"arn:aws:securityhub:us-
west-1::standards/pci-dss/v/3.2.1"}'
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 보안 표준 비활성화 또는 활성화를 참조하세요.

- 자세한 API 내용은 명령 참조BatchEnableStandards의 섹션을 참조하세요. AWS CLI

### batch-get-automation-rules

다음 코드 예시에서는 batch-get-automation-rules을 사용하는 방법을 보여 줍니다.

AWS CLI

자동화 규칙에 대한 세부 정보를 가져오려면

다음 batch-get-automation-rules 예제에서는 지정된 자동화 규칙에 대한 세부 정보를 가져옵니다. 단일 명령을 사용하여 하나 이상의 자동화 규칙에 대한 세부 정보를 가져올 수 있습니다.

```
aws securityhub batch-get-automation-rules \
```

```
    --automation-rules-arns '["arn:aws:securityhub:us-
east-1:123456789012:automation-rule/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"]'
```

출력:

```
{
    "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": []
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 자동화 규칙 보기를 참조하세요.

- 자세한 API 내용은 명령 참조BatchGetAutomationRules의 섹션을 참조하세요. AWS CLI

### batch-get-configuration-policy-associations

다음 코드 예시에서는 batch-get-configuration-policy-associations을 사용하는 방법을 보여 줍니다.

AWS CLI

대상 배치에 대한 구성 연결 세부 정보를 가져오려면

다음 batch-get-configuration-policy-associations 예제에서는 지정된 대상에 대한 연결 세부 정보를 검색합니다. 대상의 계정IDs, 조직 단위 IDs또는 루트 ID를 제공할 수 있습니다.

```
aws securityhub batch-get-configuration-policy-associations \
    --target '{"OrganizationalUnitId": "ou-6hi7-8j91kl2m"}'
```

출력:

```
{
    "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."
}
```

자세한 내용은 Security Hub 사용 설명서의 Security Hub 구성 정책 보기를 참조하세요. AWS

- 자세한 API 내용은 명령 참조BatchGetConfigurationPolicyAssociations의 섹션을 참조하세요. AWS CLI

### batch-get-security-controls

다음 코드 예시에서는 batch-get-security-controls을 사용하는 방법을 보여 줍니다.

AWS CLI

보안 제어 세부 정보를 가져오려면

다음 batch-get-security-controls 예제에서는 현재 AWS 계정 및 AWS 리전의 보안 컨트롤 ACM.1 및 IAM.1에 대한 세부 정보를 가져옵니다.

```
aws securityhub batch-get-security-controls \
    --security-control-ids '["ACM.1", "IAM.1"]'
```

출력:

```
{
    "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": {}
        }
    ]
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 컨트롤에 대한 세부 정보 보기를 참조하세요.

- 자세한 API 내용은 명령 참조BatchGetSecurityControls의 섹션을 참조하세요. AWS CLI

### batch-get-standards-control-associations

다음 코드 예시에서는 batch-get-standards-control-associations을 사용하는 방법을 보여 줍니다.

AWS CLI

제어의 활성화 상태를 가져오려면

다음 batch-get-standards-control-associations 예제에서는 지정된 컨트롤이 지정된 표준에서 활성화되었는지 여부를 식별합니다.

```
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"}]'
```

출력:

```
{
    "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"
            ]
        }
    ]
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 특정 표준에서 제어 활성화 및 비활성화를 참조하세요.

- 자세한 API 내용은 명령 참조BatchGetStandardsControlAssociations의 섹션을 참조하세요. AWS CLI

### `batch-import-findings`

다음 코드 예시에서는 `batch-import-findings`을 사용하는 방법을 보여 줍니다.

AWS CLI

결과를 업데이트하려면

다음 `batch-import-findings` 예제에서는 결과를 업데이트합니다.

```
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"
        }]'
```

출력:

```
{
    "FailedCount": 0,
    "SuccessCount": 1,
    "FailedFindings": []
}
```

자세한 내용은 AWS Security Hub 사용 설명서 의 결과를 생성하고 업데이트 BatchImportFindings 하는 데 사용을 참조하세요.

- 자세한 API 내용은 명령 참조BatchImportFindings의 섹션을 참조하세요. AWS CLI

### batch-update-automation-rules

다음 코드 예시에서는 batch-update-automation-rules을 사용하는 방법을 보여 줍니다.

AWS CLI

자동화 규칙을 업데이트하려면

다음 batch-update-automation-rules 예제에서는 지정된 자동화 규칙을 업데이트합니다. 단일 명령으로 하나 이상의 규칙을 업데이트할 수 있습니다. Security Hub 관리자 계정만 이 명령을 실행할 수 있습니다.

```
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" \
        } \
```

```
    ]'
```

출력:

```
{
    "ProcessedAutomationRules": [
        "arn:aws:securityhub:us-east-1:123456789012:automation-rule/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    ],
    "UnprocessedAutomationRules": []
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 자동화 규칙 편집을 참조하세요.

- 자세한 API 내용은 명령 참조BatchUpdateAutomationRules의 섹션을 참조하세요. AWS CLI

### batch-update-findings

다음 코드 예시에서는 batch-update-findings을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 결과를 업데이트하려면

다음 batch-update-findings 예제에서는 두 조사 결과를 업데이트하여 메모를 추가하고 심각도 레이블을 변경한 다음 해결합니다.

```
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"}'
```

출력:

```
{
```

```
    "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": []
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 [결과 업데이트를 BatchUpdateFindings 위해 사용을 참조하세요](#).

예제 2: 단축형 구문을 사용하여 결과를 업데이트하려면

다음 `batch-update-findings` 예제에서는 두 조사 결과를 업데이트하여 메모를 추가하고 심각도 레이블을 변경한 다음 단축 구문을 사용하여 해결합니다.

```
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"
```

출력:

```
{
    "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": []
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 결과 업데이트 BatchUpdateFindings 사용을 참조하세요.

- 자세한 API 내용은 명령 참조BatchUpdateFindings의 섹션을 참조하세요. AWS CLI

### batch-update-standards-control-associations

다음 코드 예시에서는 batch-update-standards-control-associations을 사용하는 방법을 보여 줍니다.

AWS CLI

활성화된 표준에서 제어의 활성화 상태를 업데이트하려면

다음 batch-update-standards-control-associations 예제에서는 지정된 표준에서 CloudTrail.1을 비활성화합니다.

```
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"}]'
```

이 명령은 성공 시 출력을 생성하지 않습니다.

자세한 내용은 AWS Security Hub 사용 설명서의 특정 표준에서 제어 활성화 및 비활성화 및 모든 표준에서 제어 활성화 및 비활성화를 참조하세요.

- 자세한 API 내용은 명령 참조BatchUpdateStandardsControlAssociations의 섹션을 참조하세요. AWS CLI

### create-action-target

다음 코드 예시에서는 create-action-target을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 작업을 생성하려면

다음 create-action-target 예제에서는 사용자 지정 작업을 생성합니다. 작업의 이름, 설명 및 식별자를 제공합니다.

```
aws securityhub create-action-target \
    --name "Send to remediation" \
    --description "Action to send the finding for remediation tracking" \
    --id "Remediation"
```

출력:

```
{
    "ActionTargetArn": "arn:aws:securityhub:us-west-1:123456789012:action/custom/
Remediation"
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 사용자 지정 작업 생성 및 CloudWatch 이벤트 규칙과 연결을 참조하세요.

- 자세한 API 내용은 명령 참조CreateActionTarget의 섹션을 참조하세요. AWS CLI

### create-automation-rule

다음 코드 예시에서는 create-automation-rule을 사용하는 방법을 보여 줍니다.

AWS CLI

자동화 규칙을 생성하려면

다음 create-automation-rule 예제에서는 현재 AWS 계정 및 AWS 리전에 자동화 규칙을 생성합니다. Security Hub는 지정된 기준에 따라 조사 결과를 필터링하고 일치하는 조사 결과에 작업을 적용합니다. Security Hub 관리자 계정만 이 명령을 실행할 수 있습니다.

```
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"
```

출력:

```
{
    "RuleArn": "arn:aws:securityhub:us-east-1:123456789012:automation-rule/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 자동화 규칙 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateAutomationRule의 섹션을 참조하세요. AWS CLI

### `create-configuration-policy`

다음 코드 예시에서는 `create-configuration-policy`을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 정책을 생성하려면

다음 create-configuration-policy 예제에서는 지정된 설정으로 구성 정책을 생성합니다.

```
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"}'
```

출력:

```
{
    "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
                                }
                            }
                        }
                    }
                ]
            }
        }
    }
}
```

자세한 내용은 Security Hub 사용 설명서의 Security Hub 구성 정책 생성 및 연결을 참조하세요. AWS

- 자세한 API 내용은 명령 참조CreateConfigurationPolicy의 섹션을 참조하세요. AWS CLI

### `create-finding-aggregator`

다음 코드 예시에서는 `create-finding-aggregator`을 사용하는 방법을 보여 줍니다.

AWS CLI

결과 집계를 활성화하려면

다음 `create-finding-aggregator` 예제에서는 결과 집계를 구성합니다. 미국 동부(버지니아)를 집계 리전으로 지정하는 미국 동부(버지니아)에서 실행됩니다. 지정된 리전만 연결하고 새 리전은 자동으로 연결하지 않음을 나타냅니다. 연결된 리전으로 미국 서부(캘리포니아 북부)와 미국 서부(오레곤)를 선택합니다.

```
aws securityhub create-finding-aggregator \
    --region us-east-1 \
    --region-linking-mode SPECIFIED_REGIONS \
    --regions us-west-1,us-west-2
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 결과 집계 활성화를 참조하세요.

- 자세한 API 내용은 명령 참조CreateFindingAggregator의 섹션을 참조하세요. AWS CLI

### create-insight

다음 코드 예시에서는 create-insight을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 인사이트를 생성하려면

다음 create-insight 예제에서는 역할과 관련된 중요한 조사 결과를 반환하는 중요한 AWS 역할 조사 결과라는 사용자 지정 인사이트를 생성합니다.

```
aws securityhub create-insight \
    --filters '{"ResourceType": [{ "Comparison": "EQUALS", "Value": "AwsIamRole"}],
 "SeverityLabel": [{"Comparison": "EQUALS", "Value": "CRITICAL"}]}' \
    --group-by-attribute "ResourceId" \
    --name "Critical role findings"
```

출력:

```
{
    "InsightArn": "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/
custom/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 사용자 지정 인사이트 관리를 참조하세요.

- 자세한 API 내용은 명령 참조CreateInsight의 섹션을 참조하세요. AWS CLI

### create-members

다음 코드 예시에서는 create-members을 사용하는 방법을 보여 줍니다.

AWS CLI

계정을 멤버 계정으로 추가하려면

다음 `create-members` 예제에서는 두 계정을 요청 관리자 계정에 멤버 계정으로 추가합니다.

```
aws securityhub create-members \
    --account-details '[{"AccountId": "123456789111"}, {"AccountId":
 "123456789222"}]'
```

출력:

```
{
    "UnprocessedAccounts": []
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 관리자 및 멤버 계정 관리를 참조하세요.

- 자세한 API 내용은 명령 참조CreateMembers의 섹션을 참조하세요. AWS CLI

### `decline-invitations`

다음 코드 예시에서는 `decline-invitations`을 사용하는 방법을 보여 줍니다.

AWS CLI

멤버 계정 초대를 거부하려면

다음 `decline-invitations` 예제에서는 지정된 관리자 계정의 멤버 계정이 되기 위한 초대를 거부합니다. 멤버 계정은 요청 계정입니다.

```
aws securityhub decline-invitations \
    --account-ids "123456789012"
```

출력:

```
{
    "UnprocessedAccounts": []
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 관리자 및 멤버 계정 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DeclineInvitations의 섹션을 참조하세요. AWS CLI

### delete-action-target

다음 코드 예시에서는 delete-action-target을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 작업을 삭제하려면

다음 delete-action-target 예제에서는 지정된 로 식별된 사용자 지정 작업을 삭제합니다 ARN.

```
aws securityhub delete-action-target \
    --action-target-arn "arn:aws:securityhub:us-west-1:123456789012:action/custom/
Remediation"
```

출력:

```
{
    "ActionTargetArn": "arn:aws:securityhub:us-west-1:123456789012:action/custom/
Remediation"
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 사용자 지정 작업 생성 및 CloudWatch 이벤트 규칙과 연결을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteActionTarget의 섹션을 참조하세요. AWS CLI

### delete-configuration-policy

다음 코드 예시에서는 delete-configuration-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 정책을 삭제하려면

다음 delete-configuration-policy 예제에서는 지정된 구성 정책을 삭제합니다.

```
aws securityhub delete-configuration-policy \
    --identifier "arn:aws:securityhub:eu-central-1:123456789012:configuration-policy/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Security Hub 사용 설명서의 Security Hub 구성 정책 삭제 및 연결 해제를 참조하세요. AWS

- 자세한 API 내용은 명령 참조DeleteConfigurationPolicy의 섹션을 참조하세요. AWS CLI

### delete-finding-aggregator

다음 코드 예시에서는 delete-finding-aggregator을 사용하는 방법을 보여 줍니다.

AWS CLI

집계 찾기를 중지하려면

다음 delete-finding-aggregator 예제에서는 집계 찾기를 중지합니다. 집계 리전인 미국 동부(버지니아)에서 실행됩니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Security Hub 사용 설명서의 결과 집계 중지를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteFindingAggregator의 섹션을 참조하세요. AWS CLI

### delete-insight

다음 코드 예시에서는 delete-insight을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 인사이트를 삭제하려면

다음 delete-insight 예제에서는 지정된 를 사용하여 사용자 지정 인사이트를 삭제합니다ARN.

```
aws securityhub delete-insight \
    --insight-arn "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/
custom/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

출력:

```
{
   "InsightArn": "arn:aws:securityhub:eu-
central-1:123456789012:insight/123456789012/custom/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111"
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 사용자 지정 인사이트 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteInsight의 섹션을 참조하세요. AWS CLI

### delete-invitations

다음 코드 예시에서는 delete-invitations을 사용하는 방법을 보여 줍니다.

AWS CLI

멤버 계정으로 초대를 삭제하려면

다음 delete-invitations 예제에서는 지정된 관리자 계정의 멤버 계정이 되기 위한 초대를 삭제합니다. 멤버 계정은 요청 계정입니다.

```
aws securityhub delete-invitations \
    --account-ids "123456789012"
```

출력:

```
{
    "UnprocessedAccounts": []
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 관리자 및 멤버 계정 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteInvitations의 섹션을 참조하세요. AWS CLI

### delete-members

다음 코드 예시에서는 delete-members을 사용하는 방법을 보여 줍니다.

AWS CLI

멤버 계정을 삭제하려면

다음 delete-members 예제에서는 요청 관리자 계정에서 지정된 멤버 계정을 삭제합니다.

```
aws securityhub delete-members \
    --account-ids "123456789111" "123456789222"
```

출력:

```
{
    "UnprocessedAccounts": []
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 관리자 및 멤버 계정 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteMembers의 섹션을 참조하세요. AWS CLI

### describe-action-targets

다음 코드 예시에서는 describe-action-targets을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 작업에 대한 세부 정보를 검색하려면

다음 describe-action-targets 예제에서는 지정된 에서 식별한 사용자 지정 작업에 대한 정보를 검색합니다ARN.

```
aws securityhub describe-action-targets \
    --action-target-arns "arn:aws:securityhub:us-west-1:123456789012:action/custom/
Remediation"
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 사용자 지정 작업 생성 및 CloudWatch 이벤트 규칙과 연결을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeActionTargets의 섹션을 참조하세요. AWS CLI

### `describe-hub`

다음 코드 예시에서는 `describe-hub`을 사용하는 방법을 보여 줍니다.

AWS CLI

허브 리소스에 대한 정보를 가져오려면

다음 `describe-hub` 예제에서는 지정된 허브 리소스의 구독 날짜를 반환합니다. 허브 리소스는 로 식별됩니다ARN.

```
aws securityhub describe-hub \
    --hub-arn "arn:aws:securityhub:us-west-1:123456789012:hub/default"
```

출력:

```
{
    "HubArn": "arn:aws:securityhub:us-west-1:123456789012:hub/default",
    "SubscribedAt": "2019-11-19T23:15:10.046Z"
}
```

자세한 내용은 AWS CloudFormation 사용 설명서의 AWS::SecurityHub::Hub를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeHub의 섹션을 참조하세요. AWS CLI

### describe-organization-configuration

다음 코드 예시에서는 describe-organization-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

조직에 대해 Security Hub가 구성된 방법을 보려면

다음 describe-organization-configuration 예제는 Security Hub에서 조직이 구성된 방식에 대한 정보를 반환합니다. 이 예제에서는 조직에서 중앙 구성을 사용합니다. Security Hub 관리자 계정만 이 명령을 실행할 수 있습니다.

```
aws securityhub describe-organization-configuration
```

출력:

```
{
    "AutoEnable": false,
    "MemberAccountLimitReached": false,
    "AutoEnableStandards": "NONE",
    "OrganizationConfiguration": {
        "ConfigurationType": "LOCAL",
        "Status": "ENABLED",
        "StatusMessage": "Central configuration has been enabled successfully"
    }
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 AWS Organizations를 사용하여 계정 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeOrganizationConfiguration의 섹션을 참조하세요. AWS CLI

### describe-products

다음 코드 예시에서는 describe-products을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 제품 통합에 대한 정보를 반환하려면

다음 describe-products 예제에서는 사용 가능한 제품 통합을 한 번에 하나씩 반환합니다.

```
aws securityhub describe-products \
    --max-results 1
```

출력:

```
{
    "NextToken": "U2FsdGVkX18vvPlOqb7RDrWRWVFBJI46MOIAb+nZmRJmR15NoRi2gm13sdQEn3O/
pq/78dGs+bKpgA+7HMPHO0qX33/zoRI+uIG/F9yLNhcOrOWzFUdy36JcXLQji3Rpnn/
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\"}}}]}"
        }
   ]
```

```
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 제품 통합 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeProducts의 섹션을 참조하세요. AWS CLI

### `describe-standards-controls`

다음 코드 예시에서는 `describe-standards-controls`을 사용하는 방법을 보여 줍니다.

AWS CLI

활성화된 표준에서 제어 목록을 요청하려면

다음 `describe-standards-controls` 예제에서는 요청자 계정의 PCI DSS 표준 구독에 있는 제어 목록을 요청합니다. 요청은 한 번에 두 개의 제어를 반환합니다.

```
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
```

출력:

```
{
    "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+eNkPoZHVl11ip5HUYQPWSWZGmftcmJiHL8JoKEsCDuaKayiPDyLK
+LiTkShveoOdvfxXCkOBaGhohIXhsIedN+LSjQV/
l7kfCfJcq4PziNC1N9xe9aq2pjlLVZnznTfSImrodT5bRNHe4fELCQq/z+5ka
+5Lzmc11axcwTd5lKgQyQqmUVoeriHZhyIiBgWKf7oNYdBVG8OEortVWvSkoUTt
+B2ThcnC7l43kI0UNxlkZ6sc64AsW"
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 컨트롤에 대한 세부 정보 보기를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeStandardsControls의 섹션을 참조하세요. AWS CLI

### describe-standards

다음 코드 예시에서는 describe-standards을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 표준 목록을 반환하려면

다음 describe-standards 예제에서는 사용 가능한 표준 목록을 반환합니다.

```
aws securityhub describe-standards
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 Security Hub 사용 설명서의 AWS Security Hub에서 보안 표준을 참조하세요. AWS

- 자세한 API 내용은 명령 참조DescribeStandards의 섹션을 참조하세요. AWS CLI

### `disable-import-findings-for-product`

다음 코드 예시에서는 `disable-import-findings-for-product`을 사용하는 방법을 보여 줍니다.

AWS CLI

제품 통합에서 결과 수신을 중지하려면

다음 `disable-import-findings-for-product` 예제에서는 지정된 제품 통합 구독에 대한 조사 결과의 흐름을 비활성화합니다.

```
aws securityhub disable-import-findings-for-product \
    --product-subscription-arn "arn:aws:securityhub:us-west-1:123456789012:product-subscription/crowdstrike/crowdstrike-falcon"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Security Hub 사용 설명서의 제품 통합 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DisableImportFindingsForProduct의 섹션을 참조하세요. AWS CLI

### `disable-organization-admin-account`

다음 코드 예시에서는 `disable-organization-admin-account`을 사용하는 방법을 보여 줍니다.

AWS CLI

Security Hub 관리자 계정을 제거하려면

다음 `disable-organization-admin-account` 예제에서는 AWS Organizations의 Security Hub 관리자 계정으로 지정된 계정의 할당을 취소합니다.

```
aws securityhub disable-organization-admin-account \
    --admin-account-id 777788889999
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Security Hub 사용 설명서의 Security Hub 관리자 계정 지정을 참조하세요. AWS

- 자세한 API 내용은 명령 참조DisableOrganizationAdminAccount의 섹션을 참조하세요. AWS CLI

### disable-security-hub

다음 코드 예시에서는 disable-security-hub을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS Security Hub를 비활성화하려면

다음 disable-security-hub 예제에서는 요청 계정의 AWS Security Hub를 비활성화합니다.

```
aws securityhub disable-security-hub
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Security Hub 사용 설명서의 Security Hub 비활성화를 참조하세요. AWS

- 자세한 API 내용은 명령 참조DisableSecurityHub의 섹션을 참조하세요. AWS CLI

### disassociate-from-administrator-account

다음 코드 예시에서는 disassociate-from-administrator-account을 사용하는 방법을 보여 줍니다.

AWS CLI

관리자 계정에서 연결을 해제하려면

다음 disassociate-from-administrator-account 예제에서는 요청 계정과 현재 관리자 계정의 연결을 해제합니다.

```
aws securityhub disassociate-from-administrator-account
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Security Hub 사용 설명서의 관리자 및 멤버 계정 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DisassociateFromAdministratorAccount의 섹션을 참조하세요. AWS CLI

### disassociate-from-master-account

다음 코드 예시에서는 disassociate-from-master-account을 사용하는 방법을 보여 줍니다.

AWS CLI

관리자 계정에서 연결을 해제하려면

다음 disassociate-from-master-account 예제에서는 요청 계정을 현재 관리자 계정과 연결 해제합니다.

```
aws securityhub disassociate-from-master-account
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Security Hub 사용 설명서의 관리자 및 멤버 계정 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DisassociateFromMasterAccount의 섹션을 참조하세요. AWS CLI

### disassociate-members

다음 코드 예시에서는 disassociate-members을 사용하는 방법을 보여 줍니다.

AWS CLI

멤버 계정 연결을 해제하려면

다음 disassociate-members 예제에서는 요청 관리자 계정에서 지정된 멤버 계정의 연결을 해제합니다.

```
aws securityhub disassociate-members  \
    --account-ids "123456789111" "123456789222"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Security Hub 사용 설명서의 관리자 및 멤버 계정 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DisassociateMembers의 섹션을 참조하세요. AWS CLI

### enable-import-findings-for-product

다음 코드 예시에서는 enable-import-findings-for-product을 사용하는 방법을 보여 줍니다.

AWS CLI

제품 통합에서 결과 수신을 시작하려면

다음 `enable-import-findings-for-product` 예제에서는 지정된 제품 통합의 결과 흐름을 활성화합니다.

```
aws securityhub enable-import-findings-for-product \
    --product-arn "arn:aws:securityhub:us-east-1:123456789333:product/crowdstrike/
crowdstrike-falcon"
```

출력:

```
{
    "ProductSubscriptionArn": "arn:aws:securityhub:us-east-1:123456789012:product-
subscription/crowdstrike/crowdstrike-falcon"
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 제품 통합 관리를 참조하세요.

- 자세한 API 내용은 명령 참조EnableImportFindingsForProduct의 섹션을 참조하세요. AWS CLI

### `enable-organization-admin-account`

다음 코드 예시에서는 `enable-organization-admin-account`을 사용하는 방법을 보여 줍니다.

AWS CLI

조직 계정을 Security Hub 관리자 계정으로 지정하려면

다음 `enable-organization-admin-account` 예제에서는 지정된 계정을 Security Hub 관리자 계정으로 지정합니다.

```
aws securityhub enable-organization-admin-account \
    --admin-account-id 777788889999
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Security Hub 사용 설명서의 Security Hub 관리자 계정 지정을 참조하세요. AWS

- 자세한 API 내용은 명령 참조EnableOrganizationAdminAccount의 섹션을 참조하세요. AWS CLI

### enable-security-hub

다음 코드 예시에서는 enable-security-hub을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS Security Hub를 활성화하려면

다음 enable-security-hub 예제에서는 요청 계정에 대한 AWS Security Hub를 활성화합니다. 기본 표준을 활성화하도록 Security Hub를 구성합니다. 허브 리소스의 경우 태그 Security에 값을 할당합니다Department.

```
aws securityhub enable-security-hub \
    --enable-default-standards \
    --tags '{"Department": "Security"}'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Security Hub 사용 설명서의 Security Hub 활성화를 참조하세요. AWS

- 자세한 API 내용은 명령 참조EnableSecurityHub의 섹션을 참조하세요. AWS CLI

### get-administrator-account

다음 코드 예시에서는 get-administrator-account을 사용하는 방법을 보여 줍니다.

AWS CLI

관리자 계정에 대한 정보를 검색하려면

다음 get-administrator-account 예제에서는 요청 계정의 관리자 계정에 대한 정보를 검색합니다.

```
aws securityhub get-administrator-account
```

출력:

```
{
   "Master": {
      "AccountId": "123456789012",
      "InvitationId": "7ab938c5d52d7904ad09f9e7c20cc4eb",
      "InvitedAt": 2020-06-01T20:21:18.042000+00:00,
```

```
        "MemberStatus": "ASSOCIATED"
    }
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 관리자 및 멤버 계정 관리를 참조하세요.

- 자세한 API 내용은 명령 참조GetAdministratorAccount의 섹션을 참조하세요. AWS CLI

### get-configuration-policy-association

다음 코드 예시에서는 get-configuration-policy-association을 사용하는 방법을 보여 줍니다.

AWS CLI

대상에 대한 구성 연결 세부 정보를 가져오려면

다음 get-configuration-policy-association 예제에서는 지정된 대상에 대한 연결 세부 정보를 검색합니다. 대상의 계정 ID, 조직 단위 ID 또는 루트 ID를 제공할 수 있습니다.

```
aws securityhub get-configuration-policy-association \
    --target '{"OrganizationalUnitId": "ou-6hi7-8j91kl2m"}'
```

출력:

```
{
    "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."
}
```

자세한 내용은 Security Hub 사용 설명서의 Security Hub 구성 정책 보기를 참조하세요. AWS

- 자세한 API 내용은 명령 참조GetConfigurationPolicyAssociation의 섹션을 참조하세요. AWS CLI

### get-configuration-policy

다음 코드 예시에서는 get-configuration-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 정책 세부 정보를 보려면

다음 get-configuration-policy 예제에서는 지정된 구성 정책에 대한 세부 정보를 검색합니다.

```
aws securityhub get-configuration-policy \
   --identifier "arn:aws:securityhub:eu-central-1:123456789012:configuration-policy/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

출력:

```
{
    "Arn": "arn:aws:securityhub:eu-central-1:123456789012:configuration-policy/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "Id": "ce5ed1e7-9639-4e2f-9313-fa87fcef944b",
    "Name": "SampleConfigurationPolicy",
    "Description": "SampleDescription",
    "UpdatedAt": "2023-11-28T20:28:04.494000+00:00",
    "CreatedAt": "2023-11-28T20:28:04.494000+00:00",
    "ConfigurationPolicy": {
        "SecurityHub": {
            "ServiceEnabled": true,
            "EnabledStandardIdentifiers": [
                "arn:aws:securityhub:eu-central-1::standards/aws-foundational-
security-best-practices/v/1.0.0",
                "arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/
v/1.2.0"
            ],
            "SecurityControlsConfiguration": {
                "DisabledSecurityControlIdentifiers": [
                    "CloudTrail.2"
                ],
                "SecurityControlCustomParameters": [
                    {
                        "SecurityControlId": "ACM.1",
                        "Parameters": {
                            "daysToExpiration": {
                                "ValueType": "CUSTOM",
                                "Value": {
                                    "Integer": 15
                                }
```

```
                            }
                        }
                    }
                ]
            }
        }
    }
}
```

자세한 내용은 Security Hub 사용 설명서의 Security Hub 구성 정책 보기를 참조하세요. AWS

- 자세한 API 내용은 명령 참조GetConfigurationPolicy의 섹션을 참조하세요. AWS CLI

### get-enabled-standards

다음 코드 예시에서는 get-enabled-standards을 사용하는 방법을 보여 줍니다.

AWS CLI

활성화된 표준에 대한 정보를 검색하려면

다음 get-enabled-standards 예제에서는 PCI DSS 표준에 대한 정보를 검색합니다.

```
aws securityhub get-enabled-standards \
    --standards-subscription-arn "arn:aws:securityhub:us-
west-1:123456789012:subscription/pci-dss/v/3.2.1"
```

출력:

```
{
    "StandardsSubscriptions": [
        {
            "StandardsArn": "arn:aws:securityhub:us-west-1::standards/pci-dss/
v/3.2.1",
            "StandardsInput": { },
            "StandardsStatus": "READY",
            "StandardsSubscriptionArn": "arn:aws:securityhub:us-
west-1:123456789012:subscription/pci-dss/v/3.2.1"
        }
    ]
}
```

자세한 내용은 Security Hub 사용 설명서의 AWS Security Hub에서 보안 표준을 참조하세요. AWS

- 자세한 API 내용은 명령 참조GetEnabledStandards의 섹션을 참조하세요. AWS CLI

### get-finding-aggregator

다음 코드 예시에서는 get-finding-aggregator을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 결과 집계 구성을 검색하려면

다음 get-finding-aggregator 예제에서는 현재 결과 집계 구성을 검색합니다.

```
aws securityhub get-finding-aggregator \
    --finding-aggregator-arn arn:aws:securityhub:us-east-1:222222222222:finding-aggregator/123e4567-e89b-12d3-a456-426652340000
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 현재 결과 집계 구성 보기를 참조하세요.

- 자세한 API 내용은 명령 참조GetFindingAggregator의 섹션을 참조하세요. AWS CLI

### get-finding-history

다음 코드 예시에서는 get-finding-history을 사용하는 방법을 보여 줍니다.

AWS CLI

조사 결과 기록을 가져오려면

다음 get-finding-history 예제는 지정된 결과에 대한 최근 90일의 기록까지 가져옵니다. 이 예제에서 결과는 조사 결과 기록에 대한 두 개의 레코드로 제한됩니다.

```
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"
```

출력:

```
{
    "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": []
        }
    ]
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 조사 결과 기록을 참조하세요.

- 자세한 API 내용은 명령 참조GetFindingHistory의 섹션을 참조하세요. AWS CLI

### `get-findings`

다음 코드 예시에서는 `get-findings`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 특정 표준에 대해 생성된 조사 결과를 반환하려면

다음 `get-findings` 예제에서는 PCI DSS 표준에 대한 결과를 반환합니다.

```
aws securityhub get-findings \
    --filters '{"GeneratorId":[{"Value": "pci-dss","Comparison":"PREFIX"}]}' \
    --max-items 1
```

출력:


```
{
    "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": "eyJOZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxfQ=="
}
```

예제 2: 워크플로 상태가 인 중요 심각도 조사 결과를 반환하려면 NOTIFIED

다음 `get-findings` 예제에서는 심각도 레이블 값이 CRITICAL 이고 워크플로 상태가 인 결과를 반환합니다NOTIFIED. 결과는 신뢰도 값을 기준으로 내림차순으로 정렬됩니다.

```
aws securityhub get-findings \
    --filters '{"SeverityLabel":[{"Value":
 "CRITICAL","Comparison":"EQUALS"}],"WorkflowStatus":
 [{"Value":"NOTIFIED","Comparison":"EQUALS"}]}' \
    --sort-criteria '{ "Field": "Confidence", "SortOrder": "desc"}' \
    --max-items 1
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 결과 필터링 및 그룹화를 참조하세요.

- 자세한 API 내용은 명령 참조GetFindings의 섹션을 참조하세요. AWS CLI

### get-insight-results

다음 코드 예시에서는 get-insight-results을 사용하는 방법을 보여 줍니다.

AWS CLI

인사이트에 대한 결과를 검색하려면

다음 get-insight-results 예제에서는 지정된 를 사용하여 인사이트에 대한 인사이트 결과 목록을 반환합니다ARN.

```
aws securityhub get-insight-results \
    --insight-arn "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/
custom/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

출력:

```
{
    "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"
            }
        ]
    }
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 인사이트 결과 및 결과 보기 및 실행을 참조하세요.

- 자세한 API 내용은 명령 참조GetInsightResults의 섹션을 참조하세요. AWS CLI

### `get-insights`

다음 코드 예시에서는 `get-insights`을 사용하는 방법을 보여 줍니다.

AWS CLI

인사이트에 대한 세부 정보를 검색하려면

다음 `get-insights` 예제에서는 지정된 를 사용하여 인사이트에 대한 구성 세부 정보를 검색합니다ARN.

```
aws securityhub get-insights \
    --insight-arns "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/
custom/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 Security Hub의 인사이트를 참조하세요. AWS

- 자세한 API 내용은 명령 참조GetInsights의 섹션을 참조하세요. AWS CLI

### get-invitations-count

다음 코드 예시에서는 get-invitations-count을 사용하는 방법을 보여 줍니다.

AWS CLI

수락되지 않은 초대 수를 검색하려면

다음 get-invitations-count 예제에서는 요청 계정이 거부했거나 응답하지 않은 초대 수를 검색합니다.

```
aws securityhub get-invitations-count
```

출력:

```
{
  "InvitationsCount": 3
```

```
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 관리자 및 멤버 계정 관리를 참조하세요.

- 자세한 API 내용은 명령 참조GetInvitationsCount의 섹션을 참조하세요. AWS CLI

### get-master-account

다음 코드 예시에서는 get-master-account을 사용하는 방법을 보여 줍니다.

AWS CLI

관리자 계정에 대한 정보를 검색하려면

다음 get-master-account 예제에서는 요청 계정의 관리자 계정에 대한 정보를 검색합니다.

```
aws securityhub get-master-account
```

출력:

```
{
   "Master": {
      "AccountId": "123456789012",
      "InvitationId": "7ab938c5d52d7904ad09f9e7c20cc4eb",
      "InvitedAt": 2020-06-01T20:21:18.042000+00:00,
      "MemberStatus": "ASSOCIATED"
   }
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 관리자 및 멤버 계정 관리를 참조하세요.

- 자세한 API 내용은 명령 참조GetMasterAccount의 섹션을 참조하세요. AWS CLI

### get-members

다음 코드 예시에서는 get-members을 사용하는 방법을 보여 줍니다.

AWS CLI

선택한 멤버 계정에 대한 정보를 검색하려면

다음 get-members 예제에서는 지정된 멤버 계정에 대한 정보를 검색합니다.

```
aws securityhub get-members \
    --account-ids "444455556666" "777788889999"
```

출력:

```
{
    "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": [ ]
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 관리자 및 멤버 계정 관리를 참조하세요.

- 자세한 API 내용은 명령 참조GetMembers의 섹션을 참조하세요. AWS CLI

### get-security-control-definition

다음 코드 예시에서는 `get-security-control-definition`을 사용하는 방법을 보여 줍니다.

AWS CLI

보안 제어 정의 세부 정보를 가져오려면

다음 `get-security-control-definition` 예제에서는 Security Hub 보안 제어에 대한 정의 세부 정보를 검색합니다. 세부 정보에는 제어 제목, 설명, 리전 가용성, 파라미터 및 기타 정보가 포함됩니다.

```
aws securityhub get-security-control-definition \
    --security-control-id ACM.1
```

출력:

```
{
    "SecurityControlDefinition": {
        "SecurityControlId": "ACM.1",
        "Title": "Imported and ACM-issued certificates should be renewed after a
 specified time period",
        "Description": "This control checks whether an AWS Certificate Manager
 (ACM) certificate is renewed within the specified time period. It checks both
 imported certificates and certificates provided by ACM. The control fails if the
 certificate isn't renewed within the specified time period. Unless you provide a
 custom parameter value for the renewal period, Security Hub uses a default value of
 30 days.",
        "RemediationUrl": "https://docs.aws.amazon.com/console/securityhub/ACM.1/
remediation",
        "SeverityRating": "MEDIUM",
        "CurrentRegionAvailability": "AVAILABLE",
        "ParameterDefinitions": {
            "daysToExpiration": {
                "Description": "Number of days within which the ACM certificate must
 be renewed",
                "ConfigurationOptions": {
                    "Integer": {
                        "DefaultValue": 30,
                        "Min": 14,
                        "Max": 365
                    }
                }
            }
        }
    }
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 사용자 지정 제어 파라미터를 참조하세요.

- 자세한 API 내용은 명령 참조GetSecurityControlDefinition의 섹션을 참조하세요. AWS CLI

### `invite-members`

다음 코드 예시에서는 `invite-members`을 사용하는 방법을 보여 줍니다.

AWS CLI

멤버 계정에 초대장을 보내려면

다음 `invite-members` 예제에서는 지정된 멤버 계정으로 초대를 보냅니다.

```
aws securityhub invite-members \
    --account-ids "123456789111" "123456789222"
```

출력:

```
{
    "UnprocessedAccounts": []
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 관리자 및 멤버 계정 관리를 참조하세요.

- 자세한 API 내용은 명령 참조InviteMembers의 섹션을 참조하세요. AWS CLI

### `list-automation-rules`

다음 코드 예시에서는 `list-automation-rules`을 사용하는 방법을 보여 줍니다.

AWS CLI

자동화 규칙 목록을 보려면

다음 `list-automation-rules` 예제에서는 AWS 계정의 자동화 규칙을 나열합니다. Security Hub 관리자 계정만 이 명령을 실행할 수 있습니다.

```
aws securityhub list-automation-rules \
    --max-results 3 \
    --next-token NULL
```

출력:

```
{
```

```
    "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"
        }
    ]
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 자동화 규칙 보기를 참조하세요.

- 자세한 API 내용은 명령 참조ListAutomationRules의 섹션을 참조하세요. AWS CLI

### list-configuration-policies

다음 코드 예시에서는 list-configuration-policies을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 정책 요약을 나열하려면

다음 list-configuration-policies 예제에서는 조직에 대한 구성 정책의 요약을 나열합니다.

```
aws securityhub list-configuration-policies \
    --max-items 3
```

출력:

```
{
    "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
        }
}
```

자세한 내용은 Security Hub 사용 설명서의 Security Hub 구성 정책 보기를 참조하세요. AWS

- 자세한 API 내용은 명령 참조ListConfigurationPolicies의 섹션을 참조하세요. AWS CLI

### `list-configuration-policy-associations`

다음 코드 예시에서는 `list-configuration-policy-associations`을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 연결을 나열하려면

다음 `list-configuration-policy-associations` 예제에서는 조직의 구성 연결 요약을 나열합니다. 응답에는 구성 정책 및 자체 관리형 동작과의 연결이 포함됩니다.

```
aws securityhub list-configuration-policy-associations \
    --association-type "APPLIED" \
    --max-items 4
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Security Hub 사용 설명서의 Security Hub 구성 정책 보기를 참조하세요. AWS

- 자세한 API 내용은 명령 참조ListConfigurationPolicyAssociations의 섹션을 참조하세요. AWS CLI

### `list-enabled-products-for-import`

다음 코드 예시에서는 `list-enabled-products-for-import`을 사용하는 방법을 보여 줍니다.

AWS CLI

활성화된 제품 통합 목록을 반환하려면

다음 `list-enabled-products-for-import` 예제에서는 현재 활성화된 제품 통합에 ARNS 대한 구독 목록을 반환합니다.

```
aws securityhub list-enabled-products-for-import
```

출력:

```
{
    "ProductSubscriptions": [ "arn:aws:securityhub:us-west-1:123456789012:product-subscription/crowdstrike/crowdstrike-falcon", "arn:aws:securityhub:us-west-1:123456789012:product-subscription/aws/securityhub" ]
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 제품 통합 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ListEnabledProductsForImport의 섹션을 참조하세요. AWS CLI

### `list-finding-aggregators`

다음 코드 예시에서는 `list-finding-aggregators`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 위젯을 나열하려면

다음 `list-finding-aggregators` 예제에서는 결과 집계 구성ARN의 를 반환합니다.

```
aws securityhub list-finding-aggregators
```

출력:

```
{
    "FindingAggregatorArn": "arn:aws:securityhub:us-east-1:222222222222:finding-aggregator/123e4567-e89b-12d3-a456-426652340000"
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 현재 결과 집계 구성 보기를 참조하세요.

- 자세한 API 내용은 명령 참조ListFindingAggregators의 섹션을 참조하세요. AWS CLI

### `list-invitations`

다음 코드 예시에서는 `list-invitations`을 사용하는 방법을 보여 줍니다.

AWS CLI

초대 목록을 표시하려면

다음 `list-invitations` 예제에서는 요청 계정으로 전송된 초대 목록을 검색합니다.

```
aws securityhub list-invitations
```

출력:

```
{
    "Invitations": [
        {
            "AccountId": "123456789012",
            "InvitationId": "7ab938c5d52d7904ad09f9e7c20cc4eb",
            "InvitedAt": 2020-06-01T20:21:18.042000+00:00,
            "MemberStatus": "ASSOCIATED"
        }
    ],
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 관리자 및 멤버 계정 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ListInvitations의 섹션을 참조하세요. AWS CLI

### `list-members`

다음 코드 예시에서는 `list-members`을 사용하는 방법을 보여 줍니다.

AWS CLI

멤버 계정 목록을 검색하려면

다음 `list-members` 예제에서는 요청 관리자 계정의 멤버 계정 목록을 반환합니다.

```
aws securityhub list-members
```

출력:

```
{
    "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
        }
    ],
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 관리자 및 멤버 계정 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ListMembers의 섹션을 참조하세요. AWS CLI

### `list-organization-admin-accounts`

다음 코드 예시에서는 `list-organization-admin-accounts`을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 Security Hub 관리자 계정을 나열하려면

다음 `list-organization-admin-accounts` 예제에서는 조직의 Security Hub 관리자 계정을 나열합니다.

```
aws securityhub list-organization-admin-accounts
```

출력:

```
{
    AdminAccounts": [
        { "AccountId": "777788889999" },
        { "Status": "ENABLED" }
    ]
}
```

자세한 내용은 Security Hub 사용 설명서의 Security Hub 관리자 계정 지정을 참조하세요. AWS

- 자세한 API 내용은 명령 참조ListOrganizationAdminAccounts의 섹션을 참조하세요. AWS CLI

### list-security-control-definitions

다음 코드 예시에서는 list-security-control-definitions을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 사용 가능한 모든 보안 제어를 나열하려면

다음 list-security-control-definitions 예제에서는 모든 Security Hub 표준에서 사용 가능한 보안 제어를 나열합니다. 이 예제에서는 결과를 세 가지 컨트롤로 제한합니다.

```
aws securityhub list-security-control-definitions \
    --max-items 3
```

출력:

```
{
    "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/
FduIGxS7ObRtAbaUt/8/RCQcg2PU0YXI20hH/GrhoOTgv+TSm0qvQVFhkJepWmqh
+NYawjocVBeos6xzn/8qnbF9IuwGg=="
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 표준에 대한 세부 정보 보기를 참조하세요.

예제 2: 특정 표준에 사용 가능한 보안 제어를 나열하려면

다음 `list-security-control-definitions` 예제에서는 CIS AWS Foundations 벤치마크 v1.4.0에 사용할 수 있는 보안 제어를 나열합니다. 이 예제에서는 결과를 세 가지 컨트롤로 제한합니다.

```
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
```

출력:

```
{
    "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": "eyJOZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAzfQ=="
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 표준에 대한 세부 정보 보기를 참조하세요.

- 자세한 API 내용은 명령 참조ListSecurityControlDefinitions의 섹션을 참조하세요. AWS CLI

### `list-standards-control-associations`

다음 코드 예시에서는 `list-standards-control-associations`을 사용하는 방법을 보여 줍니다.

AWS CLI

활성화된 각 표준에서 제어의 활성화 상태를 가져오려면

다음 `list-standards-control-associations` 예제에서는 활성화된 각 표준에서 CloudTrail.1의 활성화 상태를 나열합니다.

```
aws securityhub list-standards-control-associations \
    --security-control-id CloudTrail.1
```

출력:

```
{
    "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)."
        }
    ]
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 특정 표준에서 제어 활성화 및 비활성화를 참조하세요.

- 자세한 API 내용은 명령 참조ListStandardsControlAssociations의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 할당된 태그를 검색하려면

다음 `list-tags-for-resource` 예제에서는 지정된 허브 리소스에 할당된 태그를 반환합니다.

```
aws securityhub list-tags-for-resource \
```

```
    --resource-arn "arn:aws:securityhub:us-west-1:123456789012:hub/default"
```

출력:

```
{
    "Tags": {
        "Department" : "Operations",
        "Area" : "USMidwest"
    }
}
```

자세한 내용은 AWS CloudFormation 사용 설명서의 AWS::SecurityHub::Hub를 참조하세요.

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### start-configuration-policy-association

다음 코드 예시에서는 start-configuration-policy-association을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 구성 정책을 연결하려면

다음 start-configuration-policy-association 예제에서는 지정된 구성 정책을 지정된 조직 단위와 연결합니다. 구성은 대상 계정, 조직 단위 또는 루트와 연결될 수 있습니다.

```
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"}'
```

출력:

```
{
    "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"
}
```

자세한 내용은 Security Hub 사용 설명서의 Security Hub 구성 정책 생성 및 연결을 참조하세요. AWS

예제 2: 자체 관리형 구성 연결

다음 start-configuration-policy-association 예제에서는 자체 관리형 구성을 지정된 계정과 연결합니다.

```
aws securityhub start-configuration-policy-association \
    --configuration-policy-identifier "SELF_MANAGED_SECURITY_HUB" \
    --target '{"OrganizationalUnitId": "123456789012"}'
```

출력:

```
{
    "ConfigurationPolicyId": "SELF_MANAGED_SECURITY_HUB",
    "TargetId": "123456789012",
    "TargetType": "ACCOUNT",
    "AssociationType": "APPLIED",
    "UpdatedAt": "2023-11-29T17:40:52.468000+00:00",
    "AssociationStatus": "PENDING"
}
```

자세한 내용은 Security Hub 사용 설명서의 Security Hub 구성 정책 생성 및 연결을 참조하세요. AWS

- 자세한 API 내용은 명령 참조StartConfigurationPolicyAssociation의 섹션을 참조하세요. AWS CLI

### start-configuration-policy-disassociation

다음 코드 예시에서는 start-configuration-policy-disassociation을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 구성 정책의 연결을 해제하려면

다음 start-configuration-policy-disassociation 예제에서는 지정된 조직 단위에서 구성 정책을 연결 해제합니다. 구성은 대상 계정, 조직 단위 또는 루트에서 연결 해제될 수 있습니다.

```
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-6hi7-8j91kl2m"}'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Security Hub 사용 설명서의 계정 및 에서 구성 연결 해제OUs를 참조하세요.

예제 2: 자체 관리형 구성의 연결을 해제하려면

다음 start-configuration-policy-disassociation 예제에서는 지정된 계정에서 자체 관리형 구성을 연결 해제합니다.

```
aws securityhub start-configuration-policy-disassociation \
    --configuration-policy-identifier "SELF_MANAGED_SECURITY_HUB" \
    --target '{"AccountId": "123456789012"}'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Security Hub 사용 설명서의 계정 및 에서 구성 연결 해제OUs를 참조하세요.

- 자세한 API 내용은 명령 참조StartConfigurationPolicyDisassociation의 섹션을 참조하세요. AWS CLI

### tag-resource

다음 코드 예시에서는 tag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 태그를 할당하려면

다음 tag-resource 예제에서는 부서 및 영역 태그의 값을 지정된 허브 리소스에 할당합니다.

```
aws securityhub tag-resource \
    --resource-arn "arn:aws:securityhub:us-west-1:123456789012:hub/default" \
    --tags '{"Department":"Operations", "Area":"USMidwest"}'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS CloudFormation 사용 설명서의 AWS::SecurityHub::Hub를 참조하세요.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### untag-resource

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에서 태그 값을 제거하려면

다음 untag-resource 예제에서는 지정된 허브 리소스에서 부서 태그를 제거합니다.

```
aws securityhub untag-resource \
    --resource-arn "arn:aws:securityhub:us-west-1:123456789012:hub/default" \
    --tag-keys "Department"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS CloudFormation 사용 설명서의 AWS::SecurityHub::Hub를 참조하세요.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-action-target

다음 코드 예시에서는 update-action-target을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 작업을 업데이트하려면

다음 update-action-target 예제에서는 지정된 에 의해 식별된 사용자 지정 작업의 이름을 업데이트합니다ARN.

```
aws securityhub update-action-target \
    --action-target-arn "arn:aws:securityhub:us-west-1:123456789012:action/custom/
Remediation" \
    --name "Send to remediation"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Security Hub 사용 설명서의 사용자 지정 작업 생성 및 CloudWatch 이벤트 규칙과 연결을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateActionTarget의 섹션을 참조하세요. AWS CLI

### update-configuration-policy

다음 코드 예시에서는 update-configuration-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

구성 정책을 업데이트하려면

다음 update-configuration-policy 예제에서는 지정된 설정을 사용하도록 기존 구성 정책을 업데이트합니다.

```
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"
```

출력:

```
{
    "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
                                }
                            }
                        }
                    }
                ]
            }
        }
    }
}
```

자세한 내용은 Security Hub 사용 설명서의 Security Hub 구성 정책 업데이트를 참조하세요. AWS

- 자세한 API 내용은 명령 참조UpdateConfigurationPolicy의 섹션을 참조하세요. AWS CLI

### update-finding-aggregator

다음 코드 예시에서는 update-finding-aggregator을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 결과 집계 구성을 업데이트하려면

다음 update-finding-aggregator 예제에서는 결과 집계 구성을 선택한 리전에서 링크로 변경합니다. 집계 리전인 미국 동부(버지니아)에서 실행됩니다. 연결된 리전으로 미국 서부(캘리포니아 북부)와 미국 서부(오레곤)를 선택합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Security Hub 사용 설명서의 결과 집계 구성 업데이트를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateFindingAggregator의 섹션을 참조하세요. AWS CLI

## update-insight

다음 코드 예시에서는 update-insight을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 사용자 지정 인사이트의 필터를 변경하려면

다음 update-insight 예제에서는 사용자 지정 인사이트의 필터를 변경합니다. 업데이트된 인사이트는 AWS 역할과 관련된 심각도가 높은 결과를 찾습니다.

```
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"
```

예제 2: 사용자 지정 인사이트의 그룹화 속성을 변경하려면

다음 update-insight 예제에서는 지정된 를 사용하여 사용자 지정 인사이트의 그룹화 속성을 변경합니다ARN. 새 그룹화 속성은 리소스 ID입니다.

```
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"
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS Security Hub 사용 설명서의 사용자 지정 인사이트 관리를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateInsight의 섹션을 참조하세요. AWS CLI

### update-organization-configuration

다음 코드 예시에서는 update-organization-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

조직에 대해 Security Hub를 구성하는 방법을 업데이트하려면

다음 update-organization-configuration 예제에서는 Security Hub가 중앙 구성을 사용하여 조직을 구성하도록 지정합니다. 이 명령을 실행한 후 위임된 Security Hub 관리자는 구성 정책을 생성하고 관리하여 조직을 구성할 수 있습니다. 위임된 관리자는 이 명령을 사용하여 중앙 구성에서 로컬 구성으로 전환할 수도 있습니다. 로컬 구성이 구성 유형인 경우 위임된 관리자는 새 조직 계정에서 Security Hub 및 기본 보안 표준을 자동으로 활성화할지 여부를 선택할 수 있습니다.

```
aws securityhub update-organization-configuration \
    --no-auto-enable \
    --organization-configuration '{"ConfigurationType": "CENTRAL"}'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Security Hub 사용 설명서의 AWS Organizations 계정 관리를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateOrganizationConfiguration의 섹션을 참조하세요. AWS CLI

## update-security-control

다음 코드 예시에서는 update-security-control을 사용하는 방법을 보여 줍니다.

AWS CLI

보안 제어 속성을 업데이트하려면

다음 update-security-control 예제에서는 Security Hub 보안 제어 파라미터의 사용자 지정 값을 지정합니다.

```
aws securityhub update-security-control \
    --security-control-id ACM.1 \
    --parameters '{"daysToExpiration": {"ValueType": "CUSTOM", "Value": {"Integer":
 15}}}' \
    --last-update-reason "Internal compliance requirement"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Security Hub 사용 설명서의 사용자 지정 제어 파라미터를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateSecurityControl의 섹션을 참조하세요. AWS CLI

### update-security-hub-configuration

다음 코드 예시에서는 update-security-hub-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

Security Hub 구성을 업데이트하려면

다음 update-security-hub-configuration 예제에서는 활성화된 표준에 대한 새 제어를 자동으로 활성화하도록 Security Hub를 구성합니다.

```
aws securityhub update-security-hub-configuration \
    --auto-enable-controls
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Security Hub 사용 설명서의 새 컨트롤 자동 활성화를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateSecurityHubConfiguration의 섹션을 참조하세요. AWS CLI

### update-standards-control

다음 코드 예시에서는 update-standards-control을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 제어를 비활성화하려면

다음 update-standards-control 예제에서는 PCI.AutoScaling.1 제어를 비활성화합니다.

```
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"
```

이 명령은 출력을 생성하지 않습니다.

예제 2: 제어를 활성화하려면

다음 update-standards-control 예제에서는 PCI.AutoScaling.1 제어를 활성화합니다.

```
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"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Security Hub 사용 설명서의 개별 제어 비활성화 및 활성화를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateStandardsControl의 섹션을 참조하세요. AWS CLI

## 를 사용한 Security Lake 예제 AWS CLI

다음 코드 예제에서는 Security Lake와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `create-aws-logsource`

다음 코드 예시에서는 create-aws-logsource을 사용하는 방법을 보여 줍니다.

AWS CLI

기본적으로 지원되는 Amazon Web Service를 Amazon Security Lake 소스로 추가하려면

다음 create-aws-logsource 예제에서는 지정된 계정 및 리전에서 VPC 흐름 로그를 Security Lake 소스로 추가합니다.

```
aws securitylake create-aws-log-source \
    --sources '[{"regions": ["us-east-1"], "accounts": ["123456789012"],
 "sourceName": "SH_FINDINGS", "sourceVersion": "2.0"}]'
```

출력:

```
{
    "failed": [
        "123456789012"
    ]
}
```

자세한 내용은 Amazon Security Lake 사용 설명서의 소스로 AWS 서비스 추가를 참조하세요.

- 자세한 API 내용은 명령 참조CreateAwsLogsource의 섹션을 참조하세요. AWS CLI

### `create-custom-logsource`

다음 코드 예시에서는 `create-custom-logsource`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 소스를 Amazon Security Lake 소스로 추가하려면

다음 `create-custom-logsource` 예제에서는 사용자 지정 소스를 지정된 로그 공급자 계정과 지정된 리전에 Security Lake 소스로 추가합니다.

```
aws securitylake create-custom-log-source \
    --source-name "VPC_FLOW" \
    --event-classes '["DNS_ACTIVITY", "NETWORK_ACTIVITY"]' \
    --configuration '{"crawlerConfiguration": {"roleArn": "arn:aws:glue:eu-
west-2:123456789012:crawler/E1WG1ZNPRXT0D4"},"providerIdentity": {"principal":
 "029189416600","externalId": "123456789012"}}' --region "us-east-1"
```

출력:

```
{
    "customLogSource": {
        "attributes": {
            "crawlerArn": "arn:aws:glue:eu-west-2:123456789012:crawler/
E1WG1ZNPRXT0D4",
```

```
            "databaseArn": "arn:aws:glue:eu-west-2:123456789012:database/
E1WG1ZNPRXT0D4",
            "tableArn": "arn:aws:glue:eu-west-2:123456789012:table/E1WG1ZNPRXT0D4"
        },
        "provider": {
            "location": "DOC-EXAMPLE-BUCKET--usw2-az1--x-s3",
            "roleArn": "arn:aws:iam::123456789012:role/AmazonSecurityLake-Provider-
testCustom2-eu-west-2"
        },
        "sourceName": "testCustom2"
        "sourceVersion": "2.0"
    }
}
```

자세한 내용은 Amazon Security Lake 사용 설명서의 사용자 지정 소스 추가를 참조하세요.

- 자세한 API 내용은 명령 참조CreateCustomLogsource의 섹션을 참조하세요. AWS CLI

### `create-data-lake-exception-subscription`

다음 코드 예시에서는 `create-data-lake-exception-subscription`을 사용하는 방법을 보여 줍니다.

AWS CLI

Security Lake 예외 알림을 보내려면

다음 `create-data-lake-exception-subscription` 예제에서는 SMS 전송을 통해 Security Lake 예외 알림을 지정된 계정으로 보냅니다. 예외 메시지는 지정된 기간 동안 유지됩니다.

```
aws securitylake create-data-lake-exception-subscription \
    --notification-endpoint "123456789012" \
    --exception-time-to-live 30 \
    --subscription-protocol "sms"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Security Lake 사용 설명서의 Amazon Security Lake 문제 해결을 참조하세요.

- 자세한 API 내용은 명령 참조CreateDataLakeExceptionSubscription의 섹션을 참조하세요. AWS CLI

### create-data-lake-organization-configuration

다음 코드 예시에서는 create-data-lake-organization-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

새 조직 계정에서 Security Lake를 구성하려면

다음 create-data-lake-organization-configuration 예제에서는 Security Lake와 새 조직 계정에서 지정된 소스 이벤트 및 로그의 수집을 활성화합니다.

```
aws securitylake create-data-lake-organization-configuration \
    --auto-enable-new-account '[{"region":"us-east-1","sources":
[{"sourceName":"SH_FINDINGS","sourceVersion": "1.0"}]}]'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Security Lake 사용 설명서의 AWS Organizations를 사용하여 여러 계정 관리를 참조하세요.

- 자세한 API 내용은 명령 참조CreateDataLakeOrganizationConfiguration의 섹션을 참조하세요. AWS CLI

### create-data-lake

다음 코드 예시에서는 create-data-lake을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 여러 리전에서 데이터 레이크를 구성하려면

다음 create-data-lake 예제에서는 여러 AWS 리전에서 Amazon Security Lake를 활성화하고 데이터 레이크를 구성합니다.

```
aws securitylake create-data-lake \
    --configurations '[{"encryptionConfiguration":
 {"kmsKeyId":"S3_MANAGED_KEY"},"region":"us-east-1","lifecycleConfiguration":
 {"expiration":{"days":365},"transitions":
[{"days":60,"storageClass":"ONEZONE_IA"}]}}, {"encryptionConfiguration":
 {"kmsKeyId":"S3_MANAGED_KEY"},"region":"us-east-2","lifecycleConfiguration":
```

```
 {"expiration":{"days":365},"transitions":
[{"days":60,"storageClass":"ONEZONE_IA"}]}}]' \
    --meta-store-manager-role-arn "arn:aws:iam:us-east-1:123456789012:role/service-
role/AmazonSecurityLakeMetaStoreManager"
```

출력:

```
{
    "dataLakes": [
        {
            "createStatus": "COMPLETED",
            "dataLakeArn": "arn:aws:securitylake:us-east-1:522481757177:data-lake/
default",
            "encryptionConfiguration": {
                "kmsKeyId": "S3_MANAGED_KEY"
            },
            "lifecycleConfiguration": {
                "expiration": {
                    "days": 365
                },
                "transitions": [
                    {
                        "days": 60,
                        "storageClass": "ONEZONE_IA"
                    }
                ]
            },
            "region": "us-east-1",
            "replicationConfiguration": {
                "regions": [
                    "ap-northeast-3"
                ],
                "roleArn": "arn:aws:securitylake:ap-northeast-3:522481757177:data-
lake/default"
            },
            "s3BucketArn": "arn:aws:s3:::aws-security-data-lake-us-east-1-
gnevt6s8z7bzby8oi3uiaysbr8v2ml",
            "updateStatus": {
                "exception": {},
                "requestId": "f20a6450-d24a-4f87-a6be-1d4c075a59c2",
                "status": "INITIALIZED"
            }
        },
```

```
        {
            "createStatus": "COMPLETED",
            "dataLakeArn": "arn:aws:securitylake:us-east-2:522481757177:data-lake/
default",
            "encryptionConfiguration": {
                "kmsKeyId": "S3_MANAGED_KEY"
            },
            "lifecycleConfiguration": {
                "expiration": {
                    "days": 365
                },
                "transitions": [
                    {
                        "days": 60,
                        "storageClass": "ONEZONE_IA"
                    }
                ]
            },
            "region": "us-east-2",
            "replicationConfiguration": {
                "regions": [
                    "ap-northeast-3"
                ],
                "roleArn": "arn:aws:securitylake:ap-northeast-3:522481757177:data-
lake/default"
            },
            "s3BucketArn": "arn:aws:s3:::aws-security-data-lake-us-east-2-
cehuifzl5rwmhm6m62h7zhvtseogr9",
            "updateStatus": {
                "exception": {},
                "requestId": "f20a6450-d24a-4f87-a6be-1d4c075a59c2",
                "status": "INITIALIZED"
            }
        }
    ]
}
```

자세한 내용은 Amazon Security Lake 사용 설명서의 Amazon Security Lake 시작하기를 참조하세요.

예제 2: 단일 리전에서 데이터 레이크를 구성하려면

다음 create-data-lake 예제에서는 단일 AWS 리전에서 Amazon Security Lake를 활성화하고 데이터 레이크를 구성합니다.

```
aws securitylake create-data-lake \
    --configurations '[{"encryptionConfiguration":
 {"kmsKeyId":"1234abcd-12ab-34cd-56ef-1234567890ab"},"region":"us-
east-2","lifecycleConfiguration": {"expiration":{"days":500},"transitions":
[{"days":30,"storageClass":"GLACIER"}]}}]' \
    --meta-store-manager-role-arn "arn:aws:iam:us-east-1:123456789012:role/service-
role/AmazonSecurityLakeMetaStoreManager"
```

출력:

```
{
    "dataLakes": [
        {
            "createStatus": "COMPLETED",
            "dataLakeArn": "arn:aws:securitylake:us-east-2:522481757177:data-lake/
default",
            "encryptionConfiguration": {
                "kmsKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
            },
            "lifecycleConfiguration": {
                "expiration": {
                    "days": 500
                },
                "transitions": [
                    {
                        "days": 30,
                        "storageClass": "GLACIER"
                    }
                ]
            },
            "region": "us-east-2",
            "replicationConfiguration": {
                "regions": [
                    "ap-northeast-3"
                ],
                "roleArn": "arn:aws:securitylake:ap-northeast-3:522481757177:data-
lake/default"
            },
            "s3BucketArn": "arn:aws:s3:::aws-security-data-lake-us-east-2-
cehuifzl5rwmhm6m62h7zhvtseogr9",
```

```
            "updateStatus": {
                "exception": {},
                "requestId": "77702a53-dcbf-493e-b8ef-518e362f3003",
                "status": "INITIALIZED"
            }
        }
    ]
}
```

자세한 내용은 Amazon Security Lake 사용 설명서의 Amazon Security Lake 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조CreateDataLake의 섹션을 참조하세요. AWS CLI

### `create-subscriber-data-access`

다음 코드 예시에서는 `create-subscriber-data-access`을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 액세스 권한이 있는 구독자를 생성하려면

다음 `create-subscriber` 예제에서는 AWS 소스에 대해 지정된 구독자 자격 증명에 대해 현재 AWS 리전의 데이터에 액세스할 수 있는 구독자를 Security Lake에 생성합니다.

```
aws securitylake create-subscriber \
    --access-types "S3" \
    --sources '[{"awsLogSource": {"sourceName": "VPC_FLOW","sourceVersion":
 "2.0"}}]' \
    --subscriber-name "opensearch-s3" \
    --subscriber-identity '{"principal": "029189416600","externalId":
 "123456789012"}'
```

출력:

```
{
    "subscriber": {
        "accessTypes": [
            "S3"
        ],
        "createdAt": "2024-07-17T19:08:26.787000+00:00",
```

```
        "roleArn": "arn:aws:iam::773172568199:role/AmazonSecurityLake-896f218b-
cfba-40be-a255-8b49a65d0407",
        "s3BucketArn": "arn:aws:s3:::aws-security-data-lake-us-east-1-
um632ufwpvxkyz0bc5hkb64atycnf3",
        "sources": [
            {
                "awsLogSource": {
                    "sourceName": "VPC_FLOW",
                    "sourceVersion": "2.0"
                }
            }
        ],
        "subscriberArn": "arn:aws:securitylake:us-
east-1:773172568199:subscriber/896f218b-cfba-40be-a255-8b49a65d0407",
        "subscriberId": "896f218b-cfba-40be-a255-8b49a65d0407",
        "subscriberIdentity": {
            "externalId": "123456789012",
            "principal": "029189416600"
        },
        "subscriberName": "opensearch-s3",
        "subscriberStatus": "ACTIVE",
        "updatedAt": "2024-07-17T19:08:27.133000+00:00"
    }
}
```

자세한 내용은 Amazon Security Lake 사용 설명서의 데이터 액세스 권한이 있는 구독자 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateSubscriberDataAccess의 섹션을 참조하세요. AWS CLI

### `create-subscriber-notification`

다음 코드 예시에서는 `create-subscriber-notification`을 사용하는 방법을 보여 줍니다.

AWS CLI

구독자 알림을 생성하려면

다음 `create-subscriber-notification` 예제에서는 새 데이터가 데이터 레이크에 기록될 때 알림을 생성하도록 구독자 알림을 지정하는 방법을 보여줍니다.

```
aws securitylake create-subscriber-notification \
    --subscriber-id "12345ab8-1a34-1c34-1bd4-12345ab9012" \
```

```
    --configuration '{"httpsNotificationConfiguration":
 {"targetRoleArn":"arn:aws:iam::XXX:role/service-role/RoleName",
 "endpoint":"https://account-management.$3.$2.securitylake.aws.dev/v1/datalake"}}'
```

출력:

```
{
    "subscriberEndpoint": [
        "https://account-management.$3.$2.securitylake.aws.dev/v1/datalake"
    ]
}
```

자세한 내용은 Amazon Security Lake 사용 설명서의 구독자 관리를 참조하세요.

- 자세한 API 내용은 명령 참조CreateSubscriberNotification의 섹션을 참조하세요. AWS CLI

### `create-subscriber-query-access`

다음 코드 예시에서는 `create-subscriber-query-access`을 사용하는 방법을 보여 줍니다.

AWS CLI

쿼리 액세스 권한이 있는 구독자를 생성하려면

다음 `create-subscriber` 예제에서는 지정된 구독자 자격 증명에 대해 현재 AWS 리전에 쿼리 액세스 권한이 있는 구독자를 Security Lake에 생성합니다.

```
aws securitylake create-subscriber \
    --access-types "LAKEFORMATION" \
    --sources '[{"awsLogSource": {"sourceName": "VPC_FLOW","sourceVersion":
 "2.0"}}]' \
    --subscriber-name "opensearch-s3" \
    --subscriber-identity '{"principal": "029189416600","externalId":
 "123456789012"}'
```

출력:

```
{
    "subscriber": {
        "accessTypes": [
            "LAKEFORMATION"
```

```
        ],
        "createdAt": "2024-07-18T01:05:55.853000+00:00",
        "resourceShareArn": "arn:aws:ram:us-east-1:123456789012:resource-
share/8c31da49-c224-4f1e-bb12-37ab756d6d8a",
        "resourceShareName": "LakeFormation-V2-NAMENAMENA-123456789012",
        "sources": [
            {
                "awsLogSource": {
                    "sourceName": "VPC_FLOW",
                    "sourceVersion": "2.0"
                }
            }
        ],
        "subscriberArn": "arn:aws:securitylake:us-east-1:123456789012:subscriber/
e762aabb-ce3d-4585-beab-63474597845d",
        "subscriberId": "e762aabb-ce3d-4585-beab-63474597845d",
        "subscriberIdentity": {
            "externalId": "123456789012",
            "principal": "029189416600"
        },
        "subscriberName": "opensearch-s3",
        "subscriberStatus": "ACTIVE",
        "updatedAt": "2024-07-18T01:05:58.393000+00:00"
    }
}
```

자세한 내용은 Amazon Security Lake 사용 설명서의 쿼리 액세스 권한이 있는 구독자 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateSubscriberQueryAccess의 섹션을 참조하세요. AWS CLI

### `delete-aws-logsource`

다음 코드 예시에서는 `delete-aws-logsource`을 사용하는 방법을 보여 줍니다.

AWS CLI

기본적으로 지원되는 AWS 서비스를 제거합니다.

다음 `delete-aws-logsource` 예제에서는 지정된 계정 및 리전에서 VPC 흐름 로그를 Security Lake 소스로 삭제합니다.

```
aws securitylake delete-aws-log-source \
```

```
    --sources '[{"regions": ["us-east-1"], "accounts": ["123456789012"],
 "sourceName": "SH_FINDINGS", "sourceVersion": "2.0"}]'
```

출력:

```
{
    "failed": [
        "123456789012"
    ]
}
```

자세한 내용은 Amazon Security Lake 사용 설명서의 소스로 AWS 서비스 제거를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteAwsLogsource의 섹션을 참조하세요. AWS CLI

### `delete-custom-logsource`

다음 코드 예시에서는 `delete-custom-logsource`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 소스를 제거합니다.

다음 `delete-custom-logsource` 예제에서는 지정된 리전의 지정된 로그 공급자 계정에서 사용자 지정 소스를 삭제합니다.

```
aws securitylake delete-custom-log-source \
    --source-name "CustomSourceName"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Security Lake 사용 설명서의 사용자 지정 소스 삭제를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteCustomLogsource의 섹션을 참조하세요. AWS CLI

### `delete-data-lake-organization-configuration`

다음 코드 예시에서는 `delete-data-lake-organization-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

멤버 계정에서 자동 소스 수집을 중지하려면

다음 delete-data-lake-organization-configuration 예제에서는 조직에 가입한 새 멤버 계정에서 AWS Security Hub 조사 결과의 자동 수집을 중지합니다. 위임된 Security Lake 관리자만 이 명령을 실행할 수 있습니다. 새 멤버 계정이 데이터 레이크에 데이터를 자동으로 기여하지 못하도록 합니다.

```
aws securitylake delete-data-lake-organization-configuration \
    --auto-enable-new-account '[{"region":"us-east-1","sources":
[{"sourceName":"SH_FINDINGS"}]}]'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Security Lake 사용 설명서의 AWS Organizations를 사용하여 여러 계정 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteDataLakeOrganizationConfiguration의 섹션을 참조하세요. AWS CLI

### delete-data-lake

다음 코드 예시에서는 delete-data-lake을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 레이크를 비활성화하려면

다음 delete-data-lake 예제에서는 지정된 AWS 리전의 데이터 레이크를 비활성화합니다. 지정된 리전에서 소스는 더 이상 데이터 레이크에 데이터를 기여하지 않습니다. AWS Organizations를 사용하는 Security Lake 배포의 경우 조직의 위임된 Security Lake 관리자만 조직 내 계정의 Security Lake를 비활성화할 수 있습니다.

```
aws securitylake delete-data-lake \
    --regions "ap-northeast-1" "eu-central-1"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Security Lake 사용 설명서의 Amazon Security Lake 비활성화를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteDataLake의 섹션을 참조하세요. AWS CLI

### delete-subscriber-notification

다음 코드 예시에서는 delete-subscriber-notification을 사용하는 방법을 보여 줍니다.

AWS CLI

구독자 알림을 삭제하려면

다음 delete-subscriber-notification 예제에서는 특정 Security Lake 구독자에 대한 구독자 알림을 삭제하는 방법을 보여줍니다.

```
aws securitylake delete-subscriber-notification \
    --subscriber-id "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Security Lake 사용 설명서의 구독자 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteSubscriberNotification의 섹션을 참조하세요. AWS CLI

### delete-subscriber

다음 코드 예시에서는 delete-subscriber을 사용하는 방법을 보여 줍니다.

AWS CLI

구독자를 삭제하려면

다음 delete-subscriber 예제에서는 구독자가 Security Lake에서 데이터를 더 이상 사용하지 않도록 하려면 구독자를 제거하는 방법을 보여줍니다.

```
aws securitylake delete-subscriber \
    --subscriber-id "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Security Lake 사용 설명서의 구독자 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteSubscriber의 섹션을 참조하세요. AWS CLI

### get-data-lake-exception-subscription

다음 코드 예시에서는 get-data-lake-exception-subscription을 사용하는 방법을 보여 줍니다.

AWS CLI

예외 구독에 대한 세부 정보를 가져오려면

다음 get-data-lake-exception-subscription 예제에서는 Security Lake 예외 구독에 대한 세부 정보를 제공합니다. 이 예제에서는 지정된 AWS 계정의 사용자에게 SMS 전송을 통해 오류를 알립니다. 예외 메시지는 지정된 기간 동안 계정에 남아 있습니다. 예외 구독은 요청자의 기본 프로토콜을 통해 Security Lake 사용자에게 오류를 알립니다.

```
aws securitylake get-data-lake-exception-subscription
```

출력:

```
{
    "exceptionTimeToLive": 30,
    "notificationEndpoint": "123456789012",
    "subscriptionProtocol": "sms"
}
```

자세한 내용은 Amazon Security Lake 사용 설명서의 데이터 레이크 상태 문제 해결을 참조하세요.

- 자세한 API 내용은 명령 참조GetDataLakeExceptionSubscription의 섹션을 참조하세요. AWS CLI

### get-data-lake-organization-configuration

다음 코드 예시에서는 get-data-lake-organization-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

새 조직 계정의 구성에 대한 세부 정보를 가져오려면

다음 get-data-lake-organization-configuration 예제에서는 Amazon Security Lake에 온보딩한 후 새 조직 계정이 전송할 소스 로그에 대한 세부 정보를 검색합니다.

```
aws securitylake get-data-lake-organization-configuration
```

출력:

```
{
    "autoEnableNewAccount": [
        {
            "region": "us-east-1",
            "sources": [
                {
                    "sourceName": "VPC_FLOW",
                    "sourceVersion": "1.0"
                },
                {
                    "sourceName": "ROUTE53",
                    "sourceVersion": "1.0"
                },
                {
                    "sourceName": "SH_FINDINGS",
                    "sourceVersion": "1.0"
                }
            ]
        }
    ]
}
```

자세한 내용은 Amazon Security Lake 사용 설명서의 AWS Organizations를 사용하여 여러 계정 관리를 참조하세요.

- 자세한 API 내용은 명령 참조GetDataLakeOrganizationConfiguration의 섹션을 참조하세요. AWS CLI

### `get-data-lake-sources`

다음 코드 예시에서는 `get-data-lake-sources`을 사용하는 방법을 보여 줍니다.

AWS CLI

로그 수집 상태를 가져오려면

다음 `get-data-lake-sources` 예제에서는 현재 AWS 리전의 지정된 계정에 대한 로그 모음의 스냅샷을 가져옵니다. 계정에 Amazon Security Lake가 활성화되어 있습니다.

```
aws securitylake get-data-lake-sources \
    --accounts "123456789012"
```

출력:

```
{
    "dataLakeSources": [
        {
            "account": "123456789012",
            "sourceName": "SH_FINDINGS",
            "sourceStatuses": [
                {
                    "resource": "vpc-1234567890abcdef0",
                    "status": "COLLECTING"
                }
            ]
        },
        {
            "account": "123456789012",
            "sourceName": "VPC_FLOW",
            "sourceStatuses": [
                {
                    "resource": "vpc-1234567890abcdef0",
                    "status": "NOT_COLLECTING"
                }
            ]
        },
        {
            "account": "123456789012",
            "sourceName": "LAMBDA_EXECUTION",
            "sourceStatuses": [
                {
                    "resource": "vpc-1234567890abcdef0",
                    "status": "COLLECTING"
                }
            ]
        },
        {
            "account": "123456789012",
            "sourceName": "ROUTE53",
            "sourceStatuses": [
                {
                    "resource": "vpc-1234567890abcdef0",
```

```
                    "status": "COLLECTING"
                }
            ]
        },
        {
            "account": "123456789012",
            "sourceName": "CLOUD_TRAIL_MGMT",
            "sourceStatuses": [
                {
                    "resource": "vpc-1234567890abcdef0",
                    "status": "COLLECTING"
                }
            ]
        }
    ],
    "dataLakeArn": null
}
```

자세한 내용은 Amazon Security Lake 사용 설명서의 AWS 서비스에서 데이터 수집을 참조하세요.

- 자세한 API 내용은 명령 참조GetDataLakeSources의 섹션을 참조하세요. AWS CLI

### get-subscriber

다음 코드 예시에서는 get-subscriber을 사용하는 방법을 보여 줍니다.

AWS CLI

구독 정보를 검색하려면

다음 get-subscriber 예제에서는 지정된 Securiy Lake 구독자에 대한 구독 정보를 검색합니다.

```
aws securitylake get-subscriber \
    --subscriber-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

출력:

```
{
    "subscriber": {
        "accessTypes": [
            "LAKEFORMATION"
        ],
        "createdAt": "2024-04-19T15:19:44.421803+00:00",
```

```
        "resourceShareArn": "arn:aws:ram:eu-west-2:123456789012:resource-share/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "resourceShareName": "LakeFormation-V3-TKJGBHCKTZ-123456789012",
        "sources": [
            {
                "awsLogSource": {
                    "sourceName": "LAMBDA_EXECUTION",
                    "sourceVersion": "1.0"
                }
            },
            {
                "awsLogSource": {
                    "sourceName": "EKS_AUDIT",
                    "sourceVersion": "2.0"
                }
            },
            {
                "awsLogSource": {
                    "sourceName": "ROUTE53",
                    "sourceVersion": "1.0"
                }
            },
            {
                "awsLogSource": {
                    "sourceName": "SH_FINDINGS",
                    "sourceVersion": "1.0"
                }
            },
            {
                "awsLogSource": {
                    "sourceName": "VPC_FLOW",
                    "sourceVersion": "1.0"
                }
            },
            {
                "customLogSource": {
                    "attributes": {
                        "crawlerArn": "arn:aws:glue:eu-west-2:123456789012:crawler/
testCustom2",
                        "databaseArn": "arn:aws:glue:eu-
west-2:123456789012:database/amazon_security_lake_glue_db_eu_west_2",
                        "tableArn": "arn:aws:glue:eu-west-2:123456789012:table/
amazon_security_lake_table_eu_west_2_ext_testcustom2"
                    },
```

```
                    "provider": {
                        "location": "s3://aws-security-data-lake-eu-
west-2-8ugsus4ztnsfpjbldwbgf4vge98av9/ext/testCustom2/",
                        "roleArn": "arn:aws:iam::123456789012:role/
AmazonSecurityLake-Provider-testCustom2-eu-west-2"
                    },
                    "sourceName": "testCustom2"
                }
            },
            {
                "customLogSource": {
                    "attributes": {
                        "crawlerArn": "arn:aws:glue:eu-west-2:123456789012:crawler/
TestCustom",
                        "databaseArn": "arn:aws:glue:eu-
west-2:123456789012:database/amazon_security_lake_glue_db_eu_west_2",
                        "tableArn": "arn:aws:glue:eu-west-2:123456789012:table/
amazon_security_lake_table_eu_west_2_ext_testcustom"
                    },
                    "provider": {
                        "location": "s3://aws-security-data-lake-eu-
west-2-8ugsus4ztnsfpjbldwbgf4vge98av9/ext/TestCustom/",
                        "roleArn": "arn:aws:iam::123456789012:role/
AmazonSecurityLake-Provider-TestCustom-eu-west-2"
                    },
                    "sourceName": "TestCustom"
                }
            }
        ],
        "subscriberArn": "arn:aws:securitylake:eu-west-2:123456789012:subscriber/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "subscriberId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "subscriberIdentity": {
            "externalId": "123456789012",
            "principal": "123456789012"
        },
        "subscriberName": "test",
        "subscriberStatus": "ACTIVE",
        "updatedAt": "2024-04-19T15:19:55.230588+00:00"
    }
}
```

자세한 내용은 Amazon Security Lake 사용 설명서의 구독자 관리를 참조하세요.

- 자세한 API 내용은 명령 참조GetSubscriber의 섹션을 참조하세요. AWS CLI

### `list-data-lake-exceptions`

다음 코드 예시에서는 `list-data-lake-exceptions`을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 레이크에 영향을 미치는 문제를 나열하려면

다음 `list-data-lake-exceptions` 예제에서는 지정된 AWS 리전에서 지난 14일 동안 데이터 레이크에 영향을 미치는 문제를 나열합니다.

```
aws securitylake list-data-lake-exceptions \
    --regions "us-east-1" "eu-west-3"
```

출력:

```
{
    "exceptions": [
        {
            "exception": "The account does not have the required role permissions.
 Update your role permissions to use the new data source version.",
            "region": "us-east-1",
            "timestamp": "2024-02-29T12:24:15.641725+00:00"
        },
        {
            "exception": "The account does not have the required role permissions.
 Update your role permissions to use the new data source version.",
            "region": "eu-west-3",
            "timestamp": "2024-02-29T12:24:15.641725+00:00"
        }
    ]
}
```

자세한 내용은 Amazon Security Lake 사용 설명서의 Amazon Security Lake 문제 해결을 참조하세요.

- 자세한 API 내용은 명령 참조ListDataLakeExceptions의 섹션을 참조하세요. AWS CLI

### list-data-lakes

다음 코드 예시에서는 list-data-lakes을 사용하는 방법을 보여 줍니다.

AWS CLI

Security Lake 구성 객체를 나열하려면

다음 list-data-lakes 예제에서는 지정된 AWS 리전에 대한 Amazon Security Lake 구성 객체를 나열합니다. 이 명령을 사용하여 Security Lake가 지정된 리전 또는 리전에서 활성화되어 있는지 확인할 수 있습니다.

```
aws securitylake list-data-lakes \
    --regions "us-east-1"
```

출력:

```
{
    "dataLakes": [
        {
            "createStatus": "COMPLETED",
            "dataLakeArn": "arn:aws:securitylake:us-east-1:123456789012:data-lake/
default",
            "encryptionConfiguration": {
                "kmsKeyId": "S3_MANAGED_KEY"
            },
            "lifecycleConfiguration": {
                "expiration": {
                    "days": 365
                },
                "transitions": [
                    {
                        "days": 60,
                        "storageClass": "ONEZONE_IA"
                    }
                ]
            },
            "region": "us-east-1",
            "replicationConfiguration": {
                "regions": [
                    "ap-northeast-3"
                ],
```

```
                "roleArn": "arn:aws:securitylake:ap-northeast-3:123456789012:data-
lake/default"
            },
            "s3BucketArn": "arn:aws:s3:::aws-security-data-lake-us-
east-1-1234567890abcdef0",
            "updateStatus": {
                "exception": {
                    "code": "software.amazon.awssdk.services.s3.model.S3Exception",
                    "reason": ""
                },
                "requestId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
                "status": "FAILED"
            }
        }
    ]
}
```

자세한 내용은 Amazon Security Lake 사용 설명서의 리전 상태 확인을 참조하세요.

- 자세한 API 내용은 명령 참조ListDataLakes의 섹션을 참조하세요. AWS CLI

### `list-log-sources`

다음 코드 예시에서는 `list-log-sources`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon Security Lake 로그 소스를 검색하려면

다음 `list-log-sources` 예제에서는 지정된 계정의 Amazon Security Lake 로그 소스를 나열합니다.

```
aws securitylake list-log-sources \
    --accounts "123456789012"
```

출력:

```
{
    "account": "123456789012",
    "region": "xy-region-1",
    "sources": [
        {
                "awsLogSource": {
```

```
                "sourceName": "VPC_FLOW",
                "sourceVersion": "2.0"
            }
        },
        {
            "awsLogSource": {
                "sourceName": "SH_FINDINGS",
                "sourceVersion": "2.0"
            }
        }
    ]
}
```

자세한 내용은 Amazon Security Lake 사용 설명서의 소스 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ListLogSources의 섹션을 참조하세요. AWS CLI

### `list-subscribers`

다음 코드 예시에서는 `list-subscribers`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon Security Lake 구독자를 검색하려면

다음 `list-subscribers` 예제에서는 특정 계정의 모든 Amazon Security Lake 구독자를 나열합니다.

```
aws securitylake list-subscribers
```

출력:

```
{
    "subscribers": [
        {
            "accessTypes": [
                "S3"
            ],
            "createdAt": "2024-06-04T15:02:28.921000+00:00",
            "roleArn": "arn:aws:iam::123456789012:role/AmazonSecurityLake-
E1WG1ZNPRXT0D4",
            "s3BucketArn": "DOC-EXAMPLE-BUCKET--usw2-az1--x-s3",
            "sources": [
```

```
                {
                    "awsLogSource": {
                        "sourceName": "CLOUD_TRAIL_MGMT",
                        "sourceVersion": "2.0"
                    }
                },
                {
                    "awsLogSource": {
                        "sourceName": "LAMBDA_EXECUTION",
                        "sourceVersion": "1.0"
                    }
                },
                {
                    "customLogSource": {
                        "attributes": {
                            "crawlerArn": "arn:aws:glue:eu-
west-2:123456789012:crawler/E1WG1ZNPRXT0D4",
                            "databaseArn": "arn:aws:glue:eu-
west-2:123456789012:database/E1WG1ZNPRXT0D4",
                            "tableArn": "arn:aws:glue:eu-west-2:123456789012:table/
E1WG1ZNPRXT0D4"
                        },
                        "provider": {
                            "location": "DOC-EXAMPLE-BUCKET--usw2-az1--x-s3",
                            "roleArn": "arn:aws:iam::123456789012:role/
AmazonSecurityLake-E1WG1ZNPRXT0D4"
                        },
                        "sourceName": "testCustom2"
                    }
                }
            ],
            "subscriberArn": "arn:aws:securitylake:eu-
west-2:123456789012:subscriber/E1WG1ZNPRXT0D4",
            "subscriberEndpoint": "arn:aws:sqs:eu-
west-2:123456789012:AmazonSecurityLake-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111-Main-
Queue",
            "subscriberId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "subscriberIdentity": {
                "externalId": "ext123456789012",
                "principal": "123456789012"
            },
            "subscriberName": "Test",
            "subscriberStatus": "ACTIVE",
            "updatedAt": "2024-06-04T15:02:35.617000+00:00"
```

```
        }
    ]
}
```

자세한 내용은 Amazon Security Lake 사용 설명서의 구독자 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ListSubscribers의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

기존 리소스의 태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 지정된 Amazon Security Lake 구독자의 태그를 나열합니다. 이 예제에서는 소유자 태그 키에 연결된 태그 값이 없습니다. 이 작업을 사용하여 다른 기존 Security Lake 리소스에 대한 태그를 나열할 수도 있습니다.

```
aws securitylake list-tags-for-resource \
    --resource-arn "arn:aws:securitylake:us-
east-1:123456789012:subscriber/1234abcd-12ab-34cd-56ef-1234567890ab"
```

출력:

```
{
    "tags": [
        {
            "key": "Environment",
            "value": "Cloud"
        },
        {
            "key": "CostCenter",
            "value": "12345"
        },
        {
            "key": "Owner",
            "value": ""
        }
    ]
}
```

자세한 내용은 Amazon Security Lake 사용 설명서의 Amazon Security Lake 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### `register-data-lake-delegated-administrator`

다음 코드 예시에서는 `register-data-lake-delegated-administrator`을 사용하는 방법을 보여 줍니다.

AWS CLI

위임된 관리자를 지정하려면

다음 `register-data-lake-delegated-administrator` 예제에서는 지정된 AWS 계정을 위임된 Amazon Security Lake 관리자로 지정합니다.

```
aws securitylake register-data-lake-delegated-administrator \
    --account-id 123456789012
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Security Lake 사용 설명서의 AWS Organizations를 사용하여 여러 계정 관리를 참조하세요.

- 자세한 API 내용은 명령 참조RegisterDataLakeDelegatedAdministrator의 섹션을 참조하세요. AWS CLI

### `tag-resource`

다음 코드 예시에서는 `tag-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

기존 리소스에 태그를 추가하려면

다음 `tag-resource` 예제에서는 기존 구독자 리소스에 태그를 추가합니다. 새 리소스를 생성하고 하나 이상의 태그를 추가하려면 이 작업을 사용하지 마세요. 대신 생성하려는 리소스 유형에 적절한 생성 작업을 사용합니다.

```
aws securitylake tag-resource \
```

```
    --resource-arn "arn:aws:securitylake:us-
east-1:123456789012:subscriber/1234abcd-12ab-34cd-56ef-1234567890ab" \
    --tags key=Environment,value=Cloud
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Security Lake 사용 설명서의 Amazon Security Lake 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### untag-resource

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

기존 리소스에서 태그를 제거하려면

다음 untag-resource 예제에서는 기존 구독자 리소스에서 지정된 태그를 제거합니다.

```
aws securitylake untag-resource \
    --resource-arn "arn:aws:securitylake:us-
east-1:123456789012:subscriber/1234abcd-12ab-34cd-56ef-1234567890ab" \
    --tags Environment Owner
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Security Lake 사용 설명서의 Amazon Security Lake 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-data-lake-exception-subscription

다음 코드 예시에서는 update-data-lake-exception-subscription을 사용하는 방법을 보여 줍니다.

AWS CLI

Security Lake 예외에 대한 알림 구독을 업데이트하려면

다음 update-data-lake-exception-subscription 예제에서는 사용자에게 Security Lake 예외를 알리는 알림 구독을 업데이트합니다.

```
aws securitylake update-data-lake-exception-subscription \
    --notification-endpoint "123456789012" \
    --exception-time-to-live 30 \
    --subscription-protocol "email"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Security Lake 사용 설명서의 Amazon Security Lake 문제 해결을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateDataLakeExceptionSubscription의 섹션을 참조하세요. AWS CLI

### update-data-lake

다음 코드 예시에서는 update-data-lake을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 데이터 레이크 설정 업데이트

다음 update-data-lake 예제에서는 Amazon Security Lake 데이터 레이크의 설정을 업데이트합니다. 이 작업을 사용하여 데이터 암호화, 스토리지 및 롤업 리전 설정을 지정할 수 있습니다.

```
aws securitylake update-data-lake \
    --configurations '[{"encryptionConfiguration":
 {"kmsKeyId":"S3_MANAGED_KEY"},"region":"us-east-1","lifecycleConfiguration":
 {"expiration":{"days":365},"transitions":
[{"days":60,"storageClass":"ONEZONE_IA"}]}}, {"encryptionConfiguration":
 {"kmsKeyId":"S3_MANAGED_KEY"},"region":"us-east-2","lifecycleConfiguration":
 {"expiration":{"days":365},"transitions":
[{"days":60,"storageClass":"ONEZONE_IA"}]}}]' \
    --meta-store-manager-role-arn "arn:aws:iam:us-east-1:123456789012:role/service-
role/AmazonSecurityLakeMetaStoreManager"
```

출력:

```
{
    "dataLakes": [
```

```
        {
            "createStatus": "COMPLETED",
            "dataLakeArn": "arn:aws:securitylake:us-east-1:522481757177:data-lake/
default",
            "encryptionConfiguration": {
                "kmsKeyId": "S3_MANAGED_KEY"
            },
            "lifecycleConfiguration": {
                "expiration": {
                    "days": 365
                },
                "transitions": [
                    {
                        "days": 60,
                        "storageClass": "ONEZONE_IA"
                    }
                ]
            },
            "region": "us-east-1",
            "replicationConfiguration": {
                "regions": [
                    "ap-northeast-3"
                ],
                "roleArn": "arn:aws:securitylake:ap-northeast-3:522481757177:data-
lake/default"
            },
            "s3BucketArn": "arn:aws:s3:::aws-security-data-lake-us-east-1-
gnevt6s8z7bzby8oi3uiaysbr8v2ml",
            "updateStatus": {
                "exception": {},
                "requestId": "f20a6450-d24a-4f87-a6be-1d4c075a59c2",
                "status": "INITIALIZED"
            }
        },
        {
            "createStatus": "COMPLETED",
            "dataLakeArn": "arn:aws:securitylake:us-east-2:522481757177:data-lake/
default",
            "encryptionConfiguration": {
                "kmsKeyId": "S3_MANAGED_KEY"
            },
            "lifecycleConfiguration": {
                "expiration": {
                    "days": 365
```

```
                },
                "transitions": [
                    {
                        "days": 60,
                        "storageClass": "ONEZONE_IA"
                    }
                ]
            },
            "region": "us-east-2",
            "replicationConfiguration": {
                "regions": [
                    "ap-northeast-3"
                ],
                "roleArn": "arn:aws:securitylake:ap-northeast-3:522481757177:data-
lake/default"
            },
            "s3BucketArn": "arn:aws:s3:::aws-security-data-lake-us-east-2-
cehuifzl5rwmhm6m62h7zhvtseogr9",
            "updateStatus": {
                "exception": {},
                "requestId": "f20a6450-d24a-4f87-a6be-1d4c075a59c2",
                "status": "INITIALIZED"
            }
        }
    ]
}
```

자세한 내용은 Amazon Security Lake 사용 설명서의 Amazon Security Lake 시작하기를 참조하세요.

예제 2: 단일 리전에서 데이터 레이크를 구성하려면

다음 `create-data-lake` 예제에서는 단일 AWS 리전에서 Amazon Security Lake를 활성화하고 데이터 레이크를 구성합니다.

```
aws securitylake create-data-lake \
    --configurations '[{"encryptionConfiguration":
 {"kmsKeyId":"1234abcd-12ab-34cd-56ef-1234567890ab"},"region":"us-
east-2","lifecycleConfiguration": {"expiration":{"days":500},"transitions":
[{"days":30,"storageClass":"GLACIER"}]}}]' \
    --meta-store-manager-role-arn "arn:aws:iam:us-east-1:123456789012:role/service-
role/AmazonSecurityLakeMetaStoreManager"
```

출력:

```
{
    "dataLakes": [
        {
            "createStatus": "COMPLETED",
            "dataLakeArn": "arn:aws:securitylake:us-east-2:522481757177:data-lake/
default",
            "encryptionConfiguration": {
                "kmsKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
            },
            "lifecycleConfiguration": {
                "expiration": {
                    "days": 500
                },
                "transitions": [
                    {
                        "days": 30,
                        "storageClass": "GLACIER"
                    }
                ]
            },
            "region": "us-east-2",
            "replicationConfiguration": {
                "regions": [
                    "ap-northeast-3"
                ],
                "roleArn": "arn:aws:securitylake:ap-northeast-3:522481757177:data-
lake/default"
            },
            "s3BucketArn": "arn:aws:s3:::aws-security-data-lake-us-east-2-
cehuifzl5rwmhm6m62h7zhvtseogr9",
            "updateStatus": {
                "exception": {},
                "requestId": "77702a53-dcbf-493e-b8ef-518e362f3003",
                "status": "INITIALIZED"
            }
        }
    ]
}
```

자세한 내용은 Amazon Security Lake 사용 설명서의 Amazon Security Lake 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateDataLake의 섹션을 참조하세요. AWS CLI

### update-subscriber-notification

다음 코드 예시에서는 update-subscriber-notification을 사용하는 방법을 보여 줍니다.

AWS CLI

구독자 알림을 업데이트하려면

다음 update-subscriber-notification 예제에서는 구독자의 알림 방법을 업데이트하는 방법을 보여줍니다.

```
aws securitylake update-subscriber-notification \
    --subscriber-id "12345ab8-1a34-1c34-1bd4-12345ab9012" \
    --configuration '{"httpsNotificationConfiguration":
 {"targetRoleArn":"arn:aws:iam::XXX:role/service-role/RoleName",
 "endpoint":"https://account-management.$3.$2.securitylake.aws.dev/v1/datalake"}}'
```

출력:

```
{
    "subscriberEndpoint": [
        "https://account-management.$3.$2.securitylake.aws.dev/v1/datalake"
    ]
}
```

자세한 내용은 Amazon Security Lake 사용 설명서의 구독자 관리를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateSubscriberNotification의 섹션을 참조하세요. AWS CLI

### update-subscriber

다음 코드 예시에서는 update-subscriber을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon Security Lake 구독자를 업데이트하는 방법.

다음 update-subscriber 예제에서는 특정 Security Lake 구독자의 보안 레이크 데이터 액세스 소스를 업데이트합니다.

```
aws securitylake update-subscriber \
    --subscriber-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

출력:

```
{
    "subscriber": {
        "accessTypes": [
            "LAKEFORMATION"
        ],
        "createdAt": "2024-04-19T15:19:44.421803+00:00",
        "resourceShareArn": "arn:aws:ram:eu-west-2:123456789012:resource-share/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "resourceShareName": "LakeFormation-V3-TKJGBHCKTZ-123456789012",
        "sources": [
            {
                "awsLogSource": {
                    "sourceName": "LAMBDA_EXECUTION",
                    "sourceVersion": "1.0"
                }
            },
            {
                "awsLogSource": {
                    "sourceName": "EKS_AUDIT",
                    "sourceVersion": "2.0"
                }
            },
            {
                "awsLogSource": {
                    "sourceName": "ROUTE53",
                    "sourceVersion": "1.0"
                }
            },
            {
                "awsLogSource": {
                    "sourceName": "SH_FINDINGS",
                    "sourceVersion": "1.0"
                }
            },
            {
                "awsLogSource": {
                    "sourceName": "VPC_FLOW",
                    "sourceVersion": "1.0"
```

```
                }
            },
            {
                "customLogSource": {
                    "attributes": {
                        "crawlerArn": "arn:aws:glue:eu-west-2:123456789012:crawler/
E1WG1ZNPRXT0D4",
                        "databaseArn": "arn:aws:glue:eu-
west-2:123456789012:database/E1WG1ZNPRXT0D4",
                        "tableArn": "arn:aws:glue:eu-west-2:123456789012:table/
E1WG1ZNPRXT0D4"
                    },
                    "provider": {
                        "location": "DOC-EXAMPLE-BUCKET--usw2-az1--x-s3",
                        "roleArn": "arn:aws:iam::123456789012:role/
AmazonSecurityLake-E1WG1ZNPRXT0D4"
                    },
                    "sourceName": "testCustom2"
                }
            }
        ],
        "subscriberArn": "arn:aws:securitylake:eu-west-2:123456789012:subscriber/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "subscriberId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "subscriberIdentity": {
            "externalId": "123456789012",
            "principal": "123456789012"
        },
        "subscriberName": "test",
        "subscriberStatus": "ACTIVE",
        "updatedAt": "2024-07-18T20:47:37.098000+00:00"
    }
}
```

자세한 내용은 Amazon Security Lake 사용 설명서의 구독자 관리를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateSubscriber의 섹션을 참조하세요. AWS CLI

## AWS Serverless Application Repository 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS Serverless Application Repository.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `put-application-policy`

다음 코드 예시에서는 `put-application-policy`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 애플리케이션을 공개적으로 공유하려면

다음은 공개적으로 애플리케이션을 `put-application-policy` 공유하므로 누구나 AWS Serverless Application Repository에서 애플리케이션을 찾고 배포할 수 있습니다.

```
aws serverlessrepo put-application-policy \
    --application-id arn:aws:serverlessrepo:us-east-1:123456789012:applications/my-test-application \
    --statements Principals='*',Actions=Deploy
```

출력:

```
{
    "Statements": [
        {
            "Actions": [
                "Deploy"
            ],
            "Principals": [
                ""
            ],
            "StatementId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
```

```
        }
    ]
}
```

예제 2: 애플리케이션을 비공개로 공유하려면

다음은 애플리케이션을 비공개로 put-application-policy 공유하므로 특정 AWS 계정만 AWS 서버리스 애플리케이션 리포지토리에서 애플리케이션을 찾고 배포할 수 있습니다.

```
aws serverlessrepo put-application-policy \
    --application-id arn:aws:serverlessrepo:us-east-1:123456789012:applications/my-test-application \
    --statements Principals=111111111111,222222222222,Actions=Deploy
```

출력:

```
{
    "Statements": [
        {
            "Actions": [
                "Deploy"
            ],
            "Principals": [
                "111111111111",
                "222222222222"
            ],
            "StatementId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
        }
    ]
}
```

자세한 내용은 AWS 서버리스 애플리케이션 리포지토리 개발자 안내서의 콘솔을 통한 애플리케이션 공유를 참조하세요.

- 자세한 API 내용은 명령 참조PutApplicationPolicy의 섹션을 참조하세요. AWS CLI

## 를 사용한 서비스 카탈로그 예제 AWS CLI

다음 코드 예제에서는 Service Catalog AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `accept-portfolio-share`

다음 코드 예시에서는 `accept-portfolio-share`을 사용하는 방법을 보여 줍니다.

AWS CLI

포트폴리오 공유를 수락하려면

다음 `accept-portfolio-share` 예제에서는 지정된 포트폴리오를 공유하기 위해 다른 사용자가 한 제안을 수락합니다.

```
aws servicecatalog accept-portfolio-share \
    --portfolio-id port-2s6wuabcdefghijk
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조AcceptPortfolioShare의 섹션을 참조하세요. AWS CLI

### `associate-principal-with-portfolio`

다음 코드 예시에서는 `associate-principal-with-portfolio`을 사용하는 방법을 보여 줍니다.

AWS CLI

보안 주체를 포트폴리오에 연결하려면

다음 `associate-principal-with-portfolio` 예제에서는 사용자를 지정된 포트폴리오와 연결합니다.

```
aws servicecatalog associate-principal-with-portfolio \
    --portfolio-id port-2s6abcdefwdh4 \
    --principal-arn arn:aws:iam::123456789012:user/usertest \
    --principal-type IAM
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조AssociatePrincipalWithPortfolio의 섹션을 참조하세요. AWS CLI

### associate-product-with-portfolio

다음 코드 예시에서는 associate-product-with-portfolio을 사용하는 방법을 보여 줍니다.

AWS CLI

제품을 포트폴리오와 연결하려면

다음 associate-product-with-portfolio 예제는 지정된 제품을 지정된 포트폴리오와 연결합니다.

```
aws servicecatalog associate-product-with-portfolio
     --product-id prod-3p5abcdef3oyk
     --portfolio-id port-2s6abcdef5wdh4
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조AssociateProductWithPortfolio의 섹션을 참조하세요. AWS CLI

### associate-tag-option-with-resource

다음 코드 예시에서는 associate-tag-option-with-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

를 리소스 TagOption 와 연결하려면

다음 associate-tag-option-with-resource 예제에서는 지정된 를 지정된 리소스 TagOption 와 연결합니다.

```
aws servicecatalog associate-tag-option-with-resource \
    --resource-id port-2s6abcdq5wdh4 \
```

```
    --tag-option-id tag-p3abc2pkpz5qc
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조AssociateTagOptionWithResource의 섹션을 참조하세요. AWS CLI

### copy-product

다음 코드 예시에서는 copy-product을 사용하는 방법을 보여 줍니다.

AWS CLI

제품을 복사하려면

다음 copy-product 예제에서는 JSON 파일을 사용하여 파라미터를 전달하는 지정된 제품의 사본을 만듭니다.

```
aws servicecatalog copy-product --cli-input-json file://copy-product-input.json
```

copy-product-input.json의 콘텐츠:

```
{
    "SourceProductArn": "arn:aws:catalog:us-west-2:123456789012:product/prod-
tcabcd3syn2xy",
    "TargetProductName": "copy-of-myproduct",
    "CopyOptions": [
        "CopyTags"
    ]
}
```

출력:

```
{
    "CopyProductToken": "copyproduct-abc5defgjkdji"
}
```

- 자세한 API 내용은 명령 참조CopyProduct의 섹션을 참조하세요. AWS CLI

### create-portfolio-share

다음 코드 예시에서는 create-portfolio-share을 사용하는 방법을 보여 줍니다.

AWS CLI

포트폴리오를 계정과 공유하려면

다음 create-portfolio-share 예제는 지정된 포트폴리오를 지정된 계정과 공유합니다.

```
aws servicecatalog create-portfolio-share \
    --portfolio-id port-2s6abcdef5wdh4 \
    --account-id 794123456789
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조CreatePortfolioShare의 섹션을 참조하세요. AWS CLI

## create-portfolio

다음 코드 예시에서는 create-portfolio을 사용하는 방법을 보여 줍니다.

AWS CLI

포트폴리오를 생성하려면

다음 create-portfolio 예제에서는 포트폴리오를 생성합니다.

```
aws servicecatalog create-portfolio  \
    --provider-name my-provider \
    --display-name my-portfolio
```

출력:

```
{
    "PortfolioDetail": {
        "ProviderName": "my-provider",
        "DisplayName": "my-portfolio",
        "CreatedTime": 1571337221.555,
        "ARN": "arn:aws:catalog:us-east-2:123456789012:portfolio/
port-2s6xmplq5wdh4",
        "Id": "port-2s6xmplq5wdh4"
    }
}
```

- 자세한 API 내용은 명령 참조CreatePortfolio의 섹션을 참조하세요. AWS CLI

### create-product

다음 코드 예시에서는 create-product을 사용하는 방법을 보여 줍니다.

AWS CLI

제품을 생성하려면

다음 create-product 예제에서는 JSON 파일을 사용하여 파라미터를 전달하는 제품을 생성합니다.

```
aws servicecatalog create-product \
    --cli-input-json file://create-product-input.json
```

create-product-input.json의 콘텐츠:

```
{
    "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"
    }
}
```

출력:

```
{
    "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"
    }
}
```

- 자세한 API 내용은 명령 참조CreateProduct의 섹션을 참조하세요. AWS CLI

### `create-provisioning-artifact`

다음 코드 예시에서는 `create-provisioning-artifact`을 사용하는 방법을 보여 줍니다.

AWS CLI

프로비저닝 아티팩트를 생성하려면

다음 create-provisioning-artifact 예제에서는 JSON 파일을 사용하여 파라미터를 전달하는 프로비저닝 아티팩트를 생성합니다.

```
aws servicecatalog create-provisioning-artifact \
   --cli-input-json file://create-provisioning-artifact-input.json
```

create-provisioning-artifact-input.json의 콘텐츠:

```
{
    "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"
    }
}
```

출력:

```
{
    "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
    }
}
```

- 자세한 API 내용은 명령 참조CreateProvisioningArtifact의 섹션을 참조하세요. AWS CLI

### `create-tag-option`

다음 코드 예시에서는 `create-tag-option`을 사용하는 방법을 보여 줍니다.

AWS CLI

를 생성하려면 TagOption

다음 `create-tag-option` 예제에서는 를 생성합니다 TagOption.

```
aws servicecatalog create-tag-option
    --key 1234
    --value name
```

출력:

```
{
    "TagOptionDetail": {
    "Id": "tag-iabcdn4fzjjms",
    "Value": "name",
    "Active": true,
    "Key": "1234"
    }
}
```

- 자세한 API 내용은 명령 참조CreateTagOption의 섹션을 참조하세요. AWS CLI

### `delete-portfolio-share`

다음 코드 예시에서는 `delete-portfolio-share`을 사용하는 방법을 보여 줍니다.

AWS CLI

포트폴리오와 계정의 공유를 중지하려면

다음 `delete-portfolio-share` 예제에서는 지정된 계정과 포트폴리오 공유를 중지합니다.

```
aws servicecatalog delete-portfolio-share \
    --portfolio-id port-2s6abcdq5wdh4 \
```

```
    --account-id 123456789012
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeletePortfolioShare의 섹션을 참조하세요. AWS CLI

### delete-portfolio

다음 코드 예시에서는 delete-portfolio을 사용하는 방법을 보여 줍니다.

AWS CLI

포트폴리오를 삭제하려면

다음 delete-portfolio 예제에서는 지정된 포트폴리오를 삭제합니다.

```
aws servicecatalog delete-portfolio \
    --id port-abcdlx4gox4do
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeletePortfolio의 섹션을 참조하세요. AWS CLI

### delete-product

다음 코드 예시에서는 delete-product을 사용하는 방법을 보여 줍니다.

AWS CLI

제품을 삭제하려면

다음 delete-product 예제에서는 지정된 제품을 삭제합니다.

```
aws servicecatalog delete-product \
    --id prod-abcdcek6yhbxi
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeleteProduct의 섹션을 참조하세요. AWS CLI

### delete-provisioning-artifact

다음 코드 예시에서는 delete-provisioning-artifact을 사용하는 방법을 보여 줍니다.

AWS CLI

프로비저닝 아티팩트를 삭제하려면

다음 delete-provisioning-artifact 예제에서는 지정된 프로비저닝 아티팩트를 삭제합니다.

```
aws servicecatalog delete-provisioning-artifact \
    --product-id prod-abc2uebuplcpw \
    --provisioning-artifact-id pa-pqabcddii7ouc
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeleteProvisioningArtifact의 섹션을 참조하세요. AWS CLI

### delete-tag-option

다음 코드 예시에서는 delete-tag-option을 사용하는 방법을 보여 줍니다.

AWS CLI

를 삭제하려면 TagOption

다음 delete-tag-option 예제에서는 지정된 를 삭제합니다 TagOption.

```
aws servicecatalog delete-tag-option \
    --id tag-iabcdn4fzjjms
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeleteTagOption의 섹션을 참조하세요. AWS CLI

### describe-copy-product-status

다음 코드 예시에서는 describe-copy-product-status을 사용하는 방법을 보여 줍니다.

AWS CLI

제품 복사 작업의 상태를 설명하려면

다음 describe-copy-product-status 예제에서는 지정된 비동기 복사 제품 작업의 현재 상태를 표시합니다.

```
aws servicecatalog describe-copy-product-status \
    --copy-product-token copyproduct-znn5tf5abcd3w
```

출력:

```
{
    "CopyProductStatus": "SUCCEEDED",
    "TargetProductId": "prod-os6hog7abcdt2"
}
```

- 자세한 API 내용은 명령 참조DescribeCopyProductStatus의 섹션을 참조하세요. AWS CLI

### describe-portfolio

다음 코드 예시에서는 describe-portfolio을 사용하는 방법을 보여 줍니다.

AWS CLI

포트폴리오를 설명하려면

다음 describe-portfolio 예제에서는 지정된 포트폴리오에 대한 세부 정보를 표시합니다.

```
aws servicecatalog describe-portfolio \
    --id port-2s6abcdq5wdh4
```

출력:

```
{
    "TagOptions": [],
    "PortfolioDetail": {
        "ARN": "arn:aws:catalog:us-west-2:687558541234:portfolio/
port-2s6abcdq5wdh4",
        "Id": "port-2s6wuzyq5wdh4",
        "CreatedTime": 1571337221.555,
        "DisplayName": "my-portfolio",
        "ProviderName": "my-provider"
    },
    "Tags": []
}
```

- 자세한 API 내용은 명령 참조DescribePortfolio의 섹션을 참조하세요. AWS CLI

### describe-product-as-admin

다음 코드 예시에서는 describe-product-as-admin을 사용하는 방법을 보여 줍니다.

AWS CLI

제품을 관리자로 설명하려면

다음 describe-product-as-admin 예제에서는 관리자 권한을 사용하여 지정된 제품에 대한 세부 정보를 표시합니다.

```
aws servicecatalog describe-product-as-admin \
    --id prod-abcdcek6yhbxi
```

출력:

```
{
    "TagOptions": [],
    "ProductViewDetail": {
        "ProductARN": "arn:aws:catalog:us-west-2:687558542028:product/prod-
abcdcek6yhbxi",
        "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-abcdcek6yhbxi",
            "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeProductAsAdmin의 섹션을 참조하세요. AWS CLI

### `describe-provisioned-product`

다음 코드 예시에서는 `describe-provisioned-product`을 사용하는 방법을 보여 줍니다.

AWS CLI

프로비저닝된 제품을 설명하려면

다음 `describe-provisioned-product` 예제에서는 지정된 프로비저닝된 제품에 대한 세부 정보를 표시합니다.

```
aws servicecatalog describe-provisioned-product \
    --id pp-dpom27bm4abcd
```

출력:

```
{
    "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": []
}
```

- 자세한 API 내용은 명령 참조DescribeProvisionedProduct의 섹션을 참조하세요. AWS CLI

### describe-provisioning-artifact

다음 코드 예시에서는 describe-provisioning-artifact을 사용하는 방법을 보여 줍니다.

AWS CLI

프로비저닝 아티팩트를 설명하려면

다음 describe-provisioning-artifact 예제에서는 지정된 프로비저닝 아티팩트에 대한 세부 정보를 표시합니다.

```
aws servicecatalog describe-provisioning-artifact \
    --provisioning-artifact-id pa-pcz347abcdcfm \
    --product-id prod-abcdfz3syn2rg
```

출력:

```
{
    "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"
}
```

- 자세한 API 내용은 명령 참조DescribeProvisioningArtifact의 섹션을 참조하세요. AWS CLI

### describe-tag-option

다음 코드 예시에서는 describe-tag-option을 사용하는 방법을 보여 줍니다.

AWS CLI

를 설명하려면 TagOption

다음 describe-tag-option 예제에서는 지정된 에 대한 세부 정보를 표시합니다 TagOption.

```
aws servicecatalog describe-tag-option \
    --id tag-p3tej2abcd5qc
```

출력:

```
{
    "TagOptionDetail": {
        "Active": true,
        "Id": "tag-p3tej2abcd5qc",
        "Value": "value-3",
        "Key": "1234"
    }
}
```

- 자세한 API 내용은 명령 참조DescribeTagOption의 섹션을 참조하세요. AWS CLI

### disassociate-principal-from-portfolio

다음 코드 예시에서는 disassociate-principal-from-portfolio을 사용하는 방법을 보여 줍니다.

AWS CLI

포트폴리오에서 보안 주체 연결을 해제하려면

다음 disassociate-principal-from-portfolio 예제에서는 지정된 보안 주체를 포트폴리오에서 연결 해제합니다.

```
aws servicecatalog disassociate-principal-from-portfolio \
```

```
    --portfolio-id port-2s6abcdq5wdh4 \
    --principal-arn arn:aws:iam::123456789012:group/myendusers
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DisassociatePrincipalFromPortfolio의 섹션을 참조하세요. AWS CLI

### disassociate-product-from-portfolio

다음 코드 예시에서는 disassociate-product-from-portfolio을 사용하는 방법을 보여 줍니다.

AWS CLI

포트폴리오에서 제품의 연결을 해제하려면

다음 disassociate-product-from-portfolio 예제에서는 지정된 제품을 포트폴리오에서 연결 해제합니다.

```
aws servicecatalog disassociate-product-from-portfolio \
    --product-id prod-3p5abcdmu3oyk \
    --portfolio-id port-2s6abcdq5wdh4
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DisassociateProductFromPortfolio의 섹션을 참조하세요. AWS CLI

### disassociate-tag-option-from-resource

다음 코드 예시에서는 disassociate-tag-option-from-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 TagOption 에서 연결을 해제하려면

다음 disassociate-tag-option-from-resource 예제에서는 지정된 를 리소스TagOption에서 연결 해제합니다.

```
aws servicecatalog disassociate-tag-option-from-resource \
     --resource-id port-2s6abcdq5wdh4 \
     --tag-option-id tag-p3abc2pkpz5qc
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DisassociateTagOptionFromResource의 섹션을 참조하세요. AWS CLI

### `list-accepted-portfolio-shares`

다음 코드 예시에서는 `list-accepted-portfolio-shares`을 사용하는 방법을 보여 줍니다.

AWS CLI

허용되는 포트폴리오 공유를 나열하려면

다음 `list-accepted-portfolio-shares` 예제에서는 기본 Service Catalog 포트폴리오만 포함하여 이 계정에서 공유를 수락한 모든 포트폴리오를 나열합니다.

```
aws servicecatalog list-accepted-portfolio-shares \
    --portfolio-share-type "AWS_SERVICECATALOG"
```

출력:

```
{
    "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"
```

```
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListAcceptedPortfolioShares의 섹션을 참조하세요. AWS CLI

### `list-portfolio-access`

다음 코드 예시에서는 `list-portfolio-access`을 사용하는 방법을 보여 줍니다.

AWS CLI

포트폴리오에 액세스할 수 있는 계정을 나열하려면

다음 `list-portfolio-access` 예제에서는 지정된 포트폴리오에 액세스할 수 있는 AWS 계정을 나열합니다.

```
aws servicecatalog list-portfolio-access \
    --portfolio-id port-2s6abcdq5wdh4
```

출력:

```
{
    "AccountIds": [
        "123456789012"
    ]
}
```

- 자세한 API 내용은 명령 참조ListPortfolioAccess의 섹션을 참조하세요. AWS CLI

### `list-portfolios-for-product`

다음 코드 예시에서는 `list-portfolios-for-product`을 사용하는 방법을 보여 줍니다.

AWS CLI

제품과 연결된 포트폴리오를 나열하려면

다음 `list-portfolios-for-product` 예제에서는 지정된 제품과 연결된 포트폴리오를 나열합니다.

```
aws servicecatalog list-portfolios-for-product \
    --product-id prod-abcdfz3syn2rg
```

출력:

```
{
    "PortfolioDetails": [
        {
            "CreatedTime": 1571337221.555,
            "Id": "port-2s6abcdq5wdh4",
            "ARN": "arn:aws:catalog:us-west-2:123456789012:portfolio/
port-2s6abcdq5wdh4",
            "DisplayName": "my-portfolio",
            "ProviderName": "my-provider"
        },
        {
            "CreatedTime": 1559665256.348,
            "Id": "port-5abcd3e5st4ei",
            "ARN": "arn:aws:catalog:us-west-2:123456789012:portfolio/
port-5abcd3e5st4ei",
            "DisplayName": "test",
            "ProviderName": "provider-name"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListPortfoliosForProduct의 섹션을 참조하세요. AWS CLI

### `list-portfolios`

다음 코드 예시에서는 `list-portfolios`을 사용하는 방법을 보여 줍니다.

AWS CLI

포트폴리오를 나열하려면

다음 `list-portfolios` 예제에서는 현재 리전의 Service Catalog 포트폴리오를 나열합니다.

```
aws servicecatalog list-portfolios
```

출력:

```
{
    "PortfolioDetails": [
        {
           "CreatedTime": 1559665256.348,
           "ARN": "arn:aws:catalog:us-east-2:123456789012:portfolio/
port-5pzcxmplst4ei",
           "DisplayName": "my-portfolio",
           "Id": "port-5pzcxmplst4ei",
           "ProviderName": "my-user"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListPortfolios의 섹션을 참조하세요. AWS CLI

### `list-principals-for-portfolio`

다음 코드 예시에서는 `list-principals-for-portfolio`을 사용하는 방법을 보여 줍니다.

AWS CLI

포트폴리오의 모든 보안 주체를 나열하려면

다음 `list-principals-for-portfolio` 예제에서는 지정된 포트폴리오의 모든 보안 주체를 나열합니다.

```
aws servicecatalog list-principals-for-portfolio \
    --portfolio-id port-2s6abcdq5wdh4
```

출력:

```
{
    "Principals": [
        {
            "PrincipalARN": "arn:aws:iam::123456789012:user/usertest",
            "PrincipalType": "IAM"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListPrincipalsForPortfolio의 섹션을 참조하세요. AWS CLI

### list-provisioning-artifacts

다음 코드 예시에서는 list-provisioning-artifacts을 사용하는 방법을 보여 줍니다.

AWS CLI

제품의 모든 프로비저닝 아티팩트를 나열하려면

다음 list-provisioning-artifacts 예제에서는 지정된 제품에 대한 모든 프로비저닝 아티팩트를 나열합니다.

```
aws servicecatalog list-provisioning-artifacts \
    --product-id prod-nfi2abcdefgcpw
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListProvisioningArtifacts의 섹션을 참조하세요. AWS CLI

### list-resources-for-tag-option

다음 코드 예시에서는 list-resources-for-tag-option을 사용하는 방법을 보여 줍니다.

AWS CLI

에 연결된 리소스를 나열하려면 TagOption

다음 `list-resources-for-tag-option` 예제에서는 지정된 와 연결된 리소스를 나열합니다TagOption.

```
aws servicecatalog list-resources-for-tag-option \
    --tag-option-id tag-p3tej2abcd5qc
```

출력:

```
{
    "ResourceDetails": [
        {
            "ARN": "arn:aws:catalog:us-west-2:123456789012:product/prod-
abcdfz3syn2rg",
            "Name": "my product",
            "Description": "description",
            "CreatedTime": 1562097906.0,
            "Id": "prod-abcdfz3syn2rg"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListResourcesForTagOption의 섹션을 참조하세요. AWS CLI

### `list-tag-options`

다음 코드 예시에서는 `list-tag-options`을 사용하는 방법을 보여 줍니다.

AWS CLI

다음 `list-tag-options` 예제에서는 의 모든 값을 나열합니다TagOptions.

```
aws servicecatalog list-tag-options
```

출력:

```
{
    "TagOptionDetails": [
```

```
        {
            "Value": "newvalue",
            "Active": true,
            "Id": "tag-iabcdn4fzjjms",
            "Key": "1234"
        },
        {
            "Value": "value1",
            "Active": true,
            "Id": "tag-e3abcdvmwvrzy",
            "Key": "key"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListTagOptions의 섹션을 참조하세요. AWS CLI

### `provision-product`

다음 코드 예시에서는 `provision-product`을 사용하는 방법을 보여 줍니다.

AWS CLI

제품을 프로비저닝하려면

다음 `provision-product` 예제에서는 지정된 프로비저닝 아티팩트를 사용하여 지정된 제품을 프로비저닝합니다.

```
aws servicecatalog provision-product \
    --product-id prod-abcdfz3syn2rg \
    --provisioning-artifact-id pa-abc347pcsccfm \
    --provisioned-product-name "mytestppname3"
```

출력:

```
{
    "RecordDetail": {
        "RecordId": "rec-tfuawdabcdege",
        "CreatedTime": 1577222793.362,
        "ProvisionedProductId": "pp-abcd27bm4mldq",
        "PathId": "lpv2-abcdg3jp6t5k6",
        "RecordErrors": [],
```

```
        "ProductId": "prod-abcdfz3syn2rg",
        "UpdatedTime": 1577222793.362,
        "RecordType": "PROVISION_PRODUCT",
        "ProvisionedProductName": "mytestppname3",
        "ProvisioningArtifactId": "pa-pcz347abcdcfm",
        "RecordTags": [],
        "Status": "CREATED",
        "ProvisionedProductType": "CFN_STACK"
    }
}
```

- 자세한 API 내용은 명령 참조ProvisionProduct의 섹션을 참조하세요. AWS CLI

### `reject-portfolio-share`

다음 코드 예시에서는 `reject-portfolio-share`을 사용하는 방법을 보여 줍니다.

AWS CLI

포트폴리오 공유를 거부하려면

다음 `reject-portfolio-share` 예제에서는 지정된 포트폴리오의 포트폴리오 공유를 거부합니다.

```
aws servicecatalog reject-portfolio-share \
    --portfolio-id port-2s6wuabcdefghijk
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조RejectPortfolioShare의 섹션을 참조하세요. AWS CLI

### `scan-provisioned-products`

다음 코드 예시에서는 `scan-provisioned-products`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 프로비저닝된 제품을 모두 나열하려면

다음 `scan-provisioned-products` 예제에서는 사용 가능한 프로비저닝된 제품을 나열합니다.

```
aws servicecatalog scan-provisioned-products
```

출력:

```
{
    "ProvisionedProducts": [
        {
            "Status": "ERROR",
            "Arn": "arn:aws:servicecatalog:us-west-2:123456789012:stack/
mytestppname3/pp-abcd27bm4mldq",
            "StatusMessage": "AmazonCloudFormationException  Parameters: [KeyName]
 must have values (Service: AmazonCloudFormation; Status Code: 400; Error Code:
 ValidationError; Request ID: 5528602a-a9ef-427c-825c-f82c31b814f5)",
            "Id": "pp-abcd27bm4mldq",
            "Type": "CFN_STACK",
            "IdempotencyToken": "527c5358-2a1a-4b9e-b1b9-7293b0ddff42",
            "CreatedTime": 1577222793.358,
            "Name": "mytestppname3",
            "LastRecordId": "rec-tfuawdabcdxge"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ScanProvisionedProducts의 섹션을 참조하세요. AWS CLI

### search-products-as-admin

다음 코드 예시에서는 search-products-as-admin을 사용하는 방법을 보여 줍니다.

AWS CLI

관리자 권한이 있는 제품을 검색하려면

다음 search-products-as-admin 예제에서는 포트폴리오 ID를 필터로 사용하여 관리자 권한이 있는 제품을 검색합니다.

```
aws servicecatalog search-products-as-admin \
    --portfolio-id port-5abcd3e5st4ei
```

출력:

```
{
    "ProductViewDetails": [
        {
```

```
            "ProductViewSummary": {
                "Name": "my product",
                "Owner": "owner name",
                "Type": "CLOUD_FORMATION_TEMPLATE",
                "ProductId": "prod-abcdfz3syn2rg",
                "HasDefaultPath": false,
                "Id": "prodview-abcdmyuzv2dlu",
                "ShortDescription": "description"
            },
            "ProductARN": "arn:aws:catalog:us-west-2:123456789012:product/prod-
abcdfz3syn2rg",
            "CreatedTime": 1562097906.0,
            "Status": "CREATED"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조SearchProductsAsAdmin의 섹션을 참조하세요. AWS CLI

### `search-provisioned-products`

다음 코드 예시에서는 `search-provisioned-products`을 사용하는 방법을 보여 줍니다.

AWS CLI

프로비저닝된 제품을 검색하려면

다음 `search-provisioned-products` 예제에서는 JSON 파일을 사용하여 파라미터를 전달하여 지정된 제품 ID와 일치하는 프로비저닝된 제품을 검색합니다.

```
aws servicecatalog search-provisioned-products \
    --cli-input-json file://search-provisioned-products-input.json
```

`search-provisioned-products-input.json`의 콘텐츠:

```
{
    "Filters": {
        "SearchQuery": [
            "prod-tcjvfz3syn2rg"
        ]
    }
}
```

출력:

```
{
    "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
}
```

- 자세한 API 내용은 명령 참조SearchProvisionedProducts의 섹션을 참조하세요. AWS CLI

### update-portfolio

다음 코드 예시에서는 update-portfolio을 사용하는 방법을 보여 줍니다.

AWS CLI

포트폴리오를 업데이트하려면

다음 update-portfolio 예제에서는 지정된 포트폴리오의 이름을 업데이트합니다.

```
aws servicecatalog update-portfolio \
    --id port-5abcd3e5st4ei \
    --display-name "New portfolio name"
```

출력:

```
{
    "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": []
}
```

- 자세한 API 내용은 명령 참조UpdatePortfolio의 섹션을 참조하세요. AWS CLI

### update-product

다음 코드 예시에서는 update-product을 사용하는 방법을 보여 줍니다.

AWS CLI

제품을 업데이트하려면

다음 update-product 예제에서는 지정된 제품의 이름과 소유자를 업데이트합니다.

```
aws servicecatalog update-product \
    --id prod-os6abc7drqlt2 \
    --name "New product name" \
    --owner "Updated product owner"
```

출력:

```
{
    "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
    }
```

```
}
```

- 자세한 API 내용은 명령 참조UpdateProduct의 섹션을 참조하세요. AWS CLI

### update-provisioning-artifact

다음 코드 예시에서는 update-provisioning-artifact을 사용하는 방법을 보여 줍니다.

AWS CLI

프로비저닝 아티팩트를 업데이트하려면

다음 update-provisioning-artifact 예제에서는 JSON 파일을 사용하여 파라미터를 전달하여 지정된 프로비저닝 아티팩트의 이름과 설명을 업데이트합니다.

```
aws servicecatalog update-provisioning-artifact \
    --cli-input-json file://update-provisioning-artifact-input.json
```

update-provisioning-artifact-input.json의 콘텐츠:

```
{
    "ProductId": "prod-abcdfz3syn2rg",
    "ProvisioningArtifactId": "pa-pcz347abcdcfm",
    "Name": "updated name",
    "Description": "updated description"
}
```

출력:

```
{
    "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
    }
}
```

- 자세한 API 내용은 명령 참조UpdateProvisioningArtifact의 섹션을 참조하세요. AWS CLI

### update-tag-option

다음 코드 예시에서는 update-tag-option을 사용하는 방법을 보여 줍니다.

AWS CLI

를 업데이트하려면 TagOption

다음 update-tag-option 예제에서는 지정된 JSON 파일을 TagOption사용하여 의 값을 업데이트합니다.

```
aws servicecatalog update-tag-option --cli-input-json file://update-tag-option-input.json
```

update-tag-option-input.json의 콘텐츠:

```
{
    "Id": "tag-iabcdn4fzjjms",
    "Value": "newvalue",
    "Active": true
}
```

출력:

```
{
    "TagOptionDetail": {
        "Value": "newvalue",
        "Key": "1234",
        "Active": true,
        "Id": "tag-iabcdn4fzjjms"
    }
}
```

- 자세한 API 내용은 명령 참조UpdateTagOption의 섹션을 참조하세요. AWS CLI

# 를 사용한 Service Quotas 예제 AWS CLI

다음 코드 예제에서는 Service Quotas 와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `get-aws-default-service-quota`

다음 코드 예시에서는 `get-aws-default-service-quota`을 사용하는 방법을 보여 줍니다.

AWS CLI

기본 서비스 할당량을 설명하려면

다음 `get-aws-default-service-quota` 예제에서는 지정된 할당량에 대한 세부 정보를 표시합니다.

```
aws service-quotas get-aws-default-service-quota \
    --service-code ec2 \
    --quota-code L-1216C47A
```

출력:

```
{
    "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"
        }
    }
}
```

- 자세한 API 내용은 명령 참조GetAwsDefaultServiceQuota의 섹션을 참조하세요. AWS CLI

### get-requested-service-quota-change

다음 코드 예시에서는 get-requested-service-quota-change을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스 할당량 증가 요청을 설명하려면

다음 get-requested-service-quota-change 예제에서는 지정된 할당량 증가 요청에 대해 설명합니다.

```
aws service-quotas get-requested-service-quota-change \
    --request-id d187537d15254312a9609aa51bbf7624u7W49tP0
```

출력:

```
{
    "RequestedQuota": {
```

```
        "Id": "d187537d15254312a9609aa51bbf7624u7W49tPO",
        "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"
    }
}
```

- 자세한 API 내용은 명령 참조GetRequestedServiceQuotaChange의 섹션을 참조하세요. AWS CLI

### get-service-quota

다음 코드 예시에서는 get-service-quota을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스 할당량을 설명하려면

다음 get-service-quota 예제에서는 지정된 할당량에 대한 세부 정보를 표시합니다.

```
aws service-quotas get-service-quota \
    --service-code ec2 \
    --quota-code L-1216C47A
```

출력:

```
{
    "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"
        }
    }
}
```

- 자세한 API 내용은 명령 참조GetServiceQuota의 섹션을 참조하세요. AWS CLI

### `list-aws-default-service-quotas`

다음 코드 예시에서는 `list-aws-default-service-quotas`을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스의 기본 할당량을 나열하려면

다음 `list-aws-default-service-quotas` 예제에서는 지정된 서비스의 할당량에 대한 기본 값을 나열합니다.

```
aws service-quotas list-aws-default-service-quotas \
    --service-code xray
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListAwsDefaultServiceQuotas의 섹션을 참조하세요. AWS CLI

### `list-requested-service-quota-change-history-by-quota`

다음 코드 예시에서는 `list-requested-service-quota-change-history-by-quota`을 사용하는 방법을 보여 줍니다.

AWS CLI

할당량 증가 요청을 나열하려면

다음 `list-requested-service-quota-change-history-by-quota` 예제에서는 지정된 할당량에 대한 할당량 증가 요청을 나열합니다.

```
aws service-quotas list-requested-service-quota-change-history-by-quota \
    --service-code ec2 \
    --quota-code L-20F13EBD
```

출력:

```
{
    "RequestedQuotas": [
        {
            "Id": "d187537d15254312a9609aa51bbf7624u7W49tPO",
            "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListRequestedServiceQuotaChangeHistoryByQuota의 섹션을 참조하세요. AWS CLI

### list-requested-service-quota-change-history

다음 코드 예시에서는 list-requested-service-quota-change-history을 사용하는 방법을 보여 줍니다.

AWS CLI

할당량 증가 요청을 나열하려면

다음 list-requested-service-quota-change-history 예제에서는 지정된 서비스에 대한 할당량 증가 요청을 나열합니다.

```
aws service-quotas list-requested-service-quota-change-history \
    --service-code ec2
```

출력:

```
{
    "RequestedQuotas": [
        {
            "Id": "d187537d15254312a9609aa51bbf7624u7W49tPO",
            "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListRequestedServiceQuotaChangeHistory의 섹션을 참조하세요. AWS CLI

### list-service-quotas

다음 코드 예시에서는 list-service-quotas을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스의 할당량을 나열하려면

다음 list-service-quotas 예제에서는 의 할당량에 대한 세부 정보를 표시합니다 AWS CloudFormation.

```
aws service-quotas list-service-quotas \
    --service-code cloudformation
```

출력:

```
{
    "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
        }
    ]
```

```
}
```

- 자세한 API 내용은 명령 참조ListServiceQuotas의 섹션을 참조하세요. AWS CLI

### `list-services`

다음 코드 예시에서는 `list-services`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 서비스를 나열하려면

다음 명령은 Service Quotas 에서 사용할 수 있는 서비스를 나열합니다.

```
aws service-quotas list-services
```

출력:

```
{
    "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"
        }
    ]
}
```

--query 파라미터를 추가하여 관심 있는 정보로 디스플레이를 필터링할 수 있습니다. 다음 예제에서는 서비스 코드만 표시합니다.

```
aws service-quotas list-services \
    --query Services[*].ServiceCode
```

출력:

```
[
    "AWSCloudMap",
    "access-analyzer",
    "acm",
    "acm-pca",
    "amplify",
    "apigateway",
    "application-autoscaling",
        ...truncated...
    "xray"
]
```

- 자세한 API 내용은 명령 참조ListServices의 섹션을 참조하세요. AWS CLI

### request-service-quota-increase

다음 코드 예시에서는 request-service-quota-increase을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스 할당량 증가를 요청하려면

다음 request-service-quota-increase 예제에서는 지정된 서비스 할당량의 증가를 요청합니다.

```
aws service-quotas request-service-quota-increase \
    --service-code ec2 \
    --quota-code L-20F13EBD \
    --desired-value 2
```

출력:

```
{
```

```
    "RequestedQuota": {
        "Id": "d187537d15254312a9609aa51bbf7624u7W49tPO",
        "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"
    }
}
```

- 자세한 API 내용은 명령 참조RequestServiceQuotaIncrease의 섹션을 참조하세요. AWS CLI

## 를 사용한 Amazon SES 예제 AWS CLI

다음 코드 예제에서는 Amazon 와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다SES.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `delete-identity`

다음 코드 예시에서는 `delete-identity`을 사용하는 방법을 보여 줍니다.

AWS CLI

자격 증명을 삭제하려면

다음 예제에서는 `delete-identity` 명령을 사용하여 Amazon 로 확인된 자격 증명 목록에서 자격 증명을 삭제합니다SES.

```
aws ses delete-identity --identity user@example.com
```

확인된 자격 증명에 대한 자세한 내용은 Amazon Simple Email Service 개발자 안내서SES의 Amazon에서 이메일 주소 및 도메인 확인을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteIdentity의 섹션을 참조하세요. AWS CLI

### `get-identity-dkim-attributes`

다음 코드 예시에서는 `get-identity-dkim-attributes`을 사용하는 방법을 보여 줍니다.

AWS CLI

자격 증명 목록에 대한 Amazon SES Easy DKIM 속성을 가져오려면

다음 예제에서는 `get-identity-dkim-attributes` 명령을 사용하여 자격 증명 목록에 대한 Amazon SES Easy DKIM 속성을 검색합니다.

```
aws ses get-identity-dkim-attributes --identities "example.com" "user@example.com"
```

출력:

```
{
   "DkimAttributes": {
       "example.com": {
           "DkimTokens": [
               "EXAMPLEjcs5xoyqytjsotsijas7236gr",
               "EXAMPLEjr76cvoc6mysspnioorxsn6ep",
               "EXAMPLEkbmkqkhlm2lyz77ppkulerm4k"
           ],
           "DkimEnabled": true,
           "DkimVerificationStatus": "Success"
       },
       "user@example.com": {
```

```
            "DkimEnabled": false,
            "DkimVerificationStatus": "NotStarted"
        }
    }
}
```

확인을 위해 제출한 적이 없는 자격 증명을 사용하여 이 명령을 직접적으로 호출하는 경우 해당 자격 증명은 출력에 표시되지 않습니다.

Easy 에 대한 자세한 내용은 Amazon Simple Email Service 개발자 안내서SES의 AmazonDKIM에서 Easy를 DKIM참조하세요.

- 자세한 API 내용은 명령 참조GetIdentityDkimAttributes의 섹션을 참조하세요. AWS CLI

### `get-identity-notification-attributes`

다음 코드 예시에서는 `get-identity-notification-attributes`을 사용하는 방법을 보여 줍니다.

AWS CLI

자격 증명 목록에 대한 Amazon SES 알림 속성을 가져오려면

다음 예제에서는 `get-identity-notification-attributes` 명령을 사용하여 자격 증명 목록에 대한 Amazon SES 알림 속성을 검색합니다.

```
aws ses get-identity-notification-attributes --
identities "user1@example.com" "user2@example.com"
```

출력:

```
{
    "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
        }
```

```
    }
}
```

이 명령은 이메일 피드백 전달 상태를 반환하고, 해당하는 경우 반송, 불만 및 전송 알림을 보내는 Amazon SNS 주제의 Amazon 리소스 이름(ARNs)을 반환합니다.

확인을 위해 제출한 적이 없는 자격 증명을 사용하여 이 명령을 직접적으로 호출하는 경우 해당 자격 증명은 출력에 표시되지 않습니다.

알림에 대한 자세한 내용은 Amazon Simple Email Service 개발자 안내서의 AmazonSES에서 알림 사용을 참조하세요.

- 자세한 API 내용은 명령 참조GetIdentityNotificationAttributes의 섹션을 참조하세요. AWS CLI

### `get-identity-verification-attributes`

다음 코드 예시에서는 `get-identity-verification-attributes`을 사용하는 방법을 보여 줍니다.

AWS CLI

자격 증명 목록에 대한 Amazon SES 확인 상태를 가져오려면

다음 예제에서는 `get-identity-verification-attributes` 명령을 사용하여 자격 증명 목록에 대한 Amazon SES 확인 상태를 검색합니다.

```
aws ses get-identity-verification-attributes --
identities "user1@example.com" "user2@example.com"
```

출력:

```
{
    "VerificationAttributes": {
        "user1@example.com": {
            "VerificationStatus": "Success"
        },
        "user2@example.com": {
            "VerificationStatus": "Pending"
        }
    }
}
```

확인을 위해 제출한 적이 없는 자격 증명을 사용하여 이 명령을 직접적으로 호출하는 경우 해당 자격 증명은 출력에 표시되지 않습니다.

확인된 자격 증명에 대한 자세한 내용은 Amazon Simple Email Service 개발자 안내서SES의 Amazon에서 이메일 주소 및 도메인 확인을 참조하세요.

- 자세한 API 내용은 명령 참조GetIdentityVerificationAttributes의 섹션을 참조하세요. AWS CLI

### `get-send-quota`

다음 코드 예시에서는 `get-send-quota`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon SES 전송 한도를 가져오려면

다음 예제에서는 `get-send-quota` 명령을 사용하여 Amazon SES 전송 한도를 반환합니다.

```
aws ses get-send-quota
```

출력:

```
{
    "Max24HourSend": 200.0,
    "SentLast24Hours": 1.0,
    "MaxSendRate": 1.0
}
```

Max24HourSend 는 24시간 동안 보낼 수 있는 최대 이메일 수인 전송 할당량입니다. 발신 할당량은 롤링 기간을 반영합니다. 이메일을 보내려고 할 때마다 Amazon은 지난 24시간 동안 보낸 이메일 수를 SES 확인합니다. 보낸 이메일의 총 수가 할당량보다 낮으면 전송 요청이 수락되고 이메일이 전송됩니다.

SentLast24Hours은 지난 24시간 동안 보낸 이메일 수입니다.

MaxSendRate 는 초당 전송할 수 있는 최대 이메일 수입니다.

발신 한도는 메시지 수가 아닌 수신자 수를 기준으로 한다는 점에 유의하세요. 예를 들어, 수신자가 10명인 이메일은 전송 할당량에서 10개로 간주됩니다.

자세한 내용은 Amazon Simple Email Service 개발자 안내서의 Amazon SES 전송 한도 관리를 참조하세요.

- 자세한 API 내용은 명령 참조GetSendQuota의 섹션을 참조하세요. AWS CLI

### `get-send-statistics`

다음 코드 예시에서는 `get-send-statistics`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon SES 전송 통계를 가져오려면

다음 예제에서는 `get-send-statistics` 명령을 사용하여 Amazon SES 전송 통계를 반환합니다.

```
aws ses get-send-statistics
```

출력:

```
{
   "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
       }
   ]
}
```

그 결과 지난 2주간의 전송 활동을 나타내는 데이터 포인트 목록이 생성됩니다. 목록의 각 데이터 포인트에는 15분 간격의 통계가 포함됩니다.

이 예제에서는 지난 2주 동안 사용자가 보낸 유일한 이메일이 2개의 15분 간격 내에 속했기 때문에 두 개의 데이터 포인트만 있습니다.

자세한 내용은 Amazon Simple Email Service 개발자 안내서의 Amazon SES 사용 통계 모니터링을 참조하세요.

- 자세한 API 내용은 명령 참조GetSendStatistics의 섹션을 참조하세요. AWS CLI

### `list-identities`

다음 코드 예시에서는 `list-identities`을 사용하는 방법을 보여 줍니다.

AWS CLI

특정 AWS 계정의 모든 자격 증명(이메일 주소 및 도메인)을 나열하려면

다음 예제에서는 `list-identities` 명령을 사용하여 Amazon 에 확인을 위해 제출된 모든 자격 증명을 나열합니다SES.

```
aws ses list-identities
```

출력:

```
{
    "Identities": [
      "user@example.com",
      "example.com"
    ]
}
```

반환되는 목록에는 확인 상태(확인됨, 확인 보류 중, 실패 등)와 상관없이 모든 자격 증명이 포함됩니다.

이 예제에서는 자격 증명 유형 파라미터를 지정하지 않았으므로 이메일 주소와 도메인이 모두 반환됩니다.

확인에 대한 자세한 내용은 Amazon Simple Email Service 개발자 안내서SES의 Amazon에서 이메일 주소 및 도메인 확인을 참조하세요.

- 자세한 API 내용은 명령 참조ListIdentities의 섹션을 참조하세요. AWS CLI

### `send-email`

다음 코드 예시에서는 `send-email`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon을 사용하여 형식이 지정된 이메일을 보내려면 SES

다음 예제에서는 `send-email` 명령을 사용하여 서식이 지정된 이메일을 보냅니다.

```
aws ses send-email --from sender@example.com --destination file://destination.json
 --message file://message.json
```

출력:

```
{
    "MessageId": "EXAMPLEf3a5efcd1-51adec81-d2a4-4e3f-9fe2-5d85c1b23783-000000"
}
```

대상과 메시지는 현재 디렉터리의 .json 파일에 저장된 JSON 데이터 구조입니다. 이러한 파일은 다음과 같습니다.

`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"
        }
    }
}
```

발신자 및 수신자 이메일 주소를 사용하려는 주소로 바꿉니다. 발신자의 이메일 주소는 Amazon 로 확인해야 합니다SES. Amazon 에 대한 프로덕션 액세스 권한이 부여될 때까지 수신자가 Amazon SES 사서함 시뮬레이터가 아닌 한 각 수신자의 이메일 주소도 확인해야 SES합니다. 확인에 대한 자세한 내용은 Amazon Simple Email Service 개발자 안내서SES의 Amazon에서 이메일 주소 및 도메인 확인을 참조하세요.

출력의 Message ID는 직접적인 send-email 호출이 성공했음을 나타냅니다.

이메일을 받지 못한 경우 정크 박스를 확인해 보세요.

형식이 지정된 이메일 전송에 대한 자세한 내용은 Amazon Simple Email Service 개발자 안내서 SESAPI의 Amazon을 사용하여 형식이 지정된 이메일 전송을 참조하세요.

- 자세한 API 내용은 명령 참조SendEmail의 섹션을 참조하세요. AWS CLI

### `send-raw-email`

다음 코드 예시에서는 `send-raw-email`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon을 사용하여 원시 이메일을 보내려면 SES

다음 예제에서는 `send-raw-email` 명령을 사용하여 TXT 첨부 파일이 포함된 이메일을 보냅니다.

```
aws ses send-raw-email --raw-message file://message.json
```

출력:

```
{
    "MessageId": "EXAMPLEf3f73d99b-c63fb06f-d263-41f8-a0fb-d0dc67d56c07-000000"
}
```

원시 메시지는 `message.json` 현재 디렉터리에 이름이 지정된 파일에 저장된 JSON 데이터 구조입니다. 이는 다음을 포함합니다.

```
{
```

```
    "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--"
}
```

보시다시피 “데이터”는 attachment.txt라는 첨부 파일을 포함하여 MIME 형식의 전체 원시 이메일 콘텐츠를 포함하는 하나의 긴 문자열입니다.

sender@example.com 및 recipient@example.com을 사용하려는 주소로 바꿉니다. 발신자의 이메일 주소는 Amazon 로 확인해야 합니다SES. Amazon 에 대한 프로덕션 액세스 권한이 부여될 때까지 수신자가 Amazon SES 사서함 시뮬레이터가 아닌 한 수신자의 이메일 주소도 확인해야 SES합니다. 확인에 대한 자세한 내용은 Amazon Simple Email Service 개발자 안내서SES의 Amazon에서 이메일 주소 및 도메인 확인을 참조하세요.

출력의 메시지 ID는 send-raw-email에 대한 호출이 성공했음을 나타냅니다.

이메일을 받지 못한 경우 정크 박스를 확인해 보세요.

원시 이메일 전송에 대한 자세한 내용은 Amazon Simple Email Service 개발자 안내서SESAPI의 Amazon을 사용하여 원시 이메일 전송을 참조하세요.

- 자세한 API 내용은 명령 참조SendRawEmail의 섹션을 참조하세요. AWS CLI

### `set-identity-dkim-enabled`

다음 코드 예시에서는 `set-identity-dkim-enabled`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon에서 SES 확인한 자격 증명DKIM에 대해 Easy를 활성화 또는 비활성화하려면

다음 예제에서는 `set-identity-dkim-enabled` 명령을 사용하여 확인된 이메일 주소에 DKIM 대해 를 비활성화합니다.

```
aws ses set-identity-dkim-enabled --identity user@example.com --no-dkim-enabled
```

Easy 에 대한 자세한 내용은 Amazon Simple Email Service 개발자 안내서SES의 AmazonDKIM에서 Easy를 DKIM참조하세요.

- 자세한 API 내용은 명령 참조SetIdentityDkimEnabled의 섹션을 참조하세요. AWS CLI

### set-identity-feedback-forwarding-enabled

다음 코드 예시에서는 set-identity-feedback-forwarding-enabled을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon SES 확인 자격 증명에 대한 반송 및 불만 이메일 피드백 전달을 활성화 또는 비활성화하려면

다음 예제에서는 set-identity-feedback-forwarding-enabled 명령을 사용하여 확인된 이메일 주소가 이메일로 반송 및 불만 사항 알림을 수신할 수 있도록 합니다.

```
aws ses set-identity-feedback-forwarding-enabled --identity user@example.com --
forwarding-enabled
```

Amazon SNS 또는 이메일 피드백 전달을 통해 반송 및 불만 사항 알림을 수신해야 하므로 반송 및 불만 사항 알림 모두에 대해 Amazon SNS 주제를 선택한 경우에만 이메일 피드백 전달을 비활성화할 수 있습니다.

알림에 대한 자세한 내용은 Amazon Simple Email Service 개발자 안내서의 AmazonSES에서 알림 사용을 참조하세요.

- 자세한 API 내용은 명령 참조SetIdentityFeedbackForwardingEnabled의 섹션을 참조하세요. AWS CLI

### set-identity-notification-topic

다음 코드 예시에서는 set-identity-notification-topic을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon이 확인된 자격 증명에 대한 반송 메일, 불만 사항 및/또는 전송 알림을 SES 게시할 Amazon SNS 주제를 설정하려면

다음 예제에서는 set-identity-notification-topic 명령을 사용하여 확인된 이메일 주소가 반송 알림을 수신할 Amazon SNS 주제를 지정합니다.

```
aws ses set-identity-notification-topic --identity user@example.com --notification-
type Bounce --sns-topic arn:aws:sns:us-east-1:EXAMPLE65304:MyTopic
```

알림에 대한 자세한 내용은 Amazon Simple Email Service 개발자 안내서의 AmazonSES에서 알림 사용을 참조하세요.

- 자세한 API 내용은 명령 참조SetIdentityNotificationTopic의 섹션을 참조하세요. AWS CLI

### `verify-domain-dkim`

다음 코드 예시에서는 `verify-domain-dkim`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon에 DKIM 서명하기 위해 확인된 도메인의 DKIM 토큰을 생성하려면 SES

다음 예제에서는 `verify-domain-dkim` 명령을 사용하여 Amazon 로 확인된 도메인에 대한 DKIM 토큰을 생성합니다SES.

```
aws ses verify-domain-dkim --domain example.com
```

출력:

```
{
   "DkimTokens": [
       "EXAMPLEq76owjnks3lnluwg65scbemvw",
       "EXAMPLEi3dnsj67hstzaj673klariwx2",
       "EXAMPLEwfbtcukvimehexktmdtaz6naj"
   ]
}
```

를 설정하려면 반환된 DKIM 토큰을 사용하여 Amazon 에서 호스팅하는 DKIM 퍼블릭 키를 가리키는 CNAME 레코드로 도메인의 DNS 설정을 업데이트해야 DKIM합니다SES. 자세한 내용은 Amazon Simple Email Service 개발자 안내서SES의 AmazonDKIM에서 Easy를 참조하세요.

- 자세한 API 내용은 명령 참조VerifyDomainDkim의 섹션을 참조하세요. AWS CLI

### `verify-domain-identity`

다음 코드 예시에서는 `verify-domain-identity`을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon을 사용하여 도메인을 확인하려면 SES

다음 예제에서는 verify-domain-identity 명령을 사용하여 도메인을 확인합니다.

```
aws ses verify-domain-identity --domain example.com
```

출력:

```
{
   "VerificationToken": "eoEmxw+YaYhb3h3iVJHuXMJXqeu1q1/wwmvjuEXAMPLE"
}
```

도메인 확인을 완료하려면 반환된 확인 토큰이 포함된 TXT 레코드를 도메인 DNS 설정에 추가해야 합니다. 자세한 내용은 Amazon Simple Email Service 개발자 안내서SES의 Amazon에서 도메인 확인을 참조하세요.

- 자세한 API 내용은 명령 참조VerifyDomainIdentity의 섹션을 참조하세요. AWS CLI

### verify-email-identity

다음 코드 예시에서는 verify-email-identity을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon에서 이메일 주소를 확인하려면 SES

다음 예제에서는 verify-email-identity 명령을 사용하여 이메일 주소를 확인합니다.

```
aws ses verify-email-identity --email-address user@example.com
```

Amazon 를 사용하여 이메일을 전송하려면 먼저 이메일을 보내는 주소 또는 도메인을 확인하여 이메일을 소유하고 있음을 증명해야 SES합니다. 아직 프로덕션 액세스 권한이 없는 경우 Amazon SES 사서함 시뮬레이터에서 제공한 이메일 주소를 제외하고 이메일을 보내는 이메일 주소도 확인해야 합니다.

가 호출되면 verify-email-identity 이메일 주소에 확인 이메일이 전송됩니다. 확인 프로세스를 완료하려면 이메일에 포함된 링크를 클릭해야 합니다.

자세한 내용은 Amazon Simple Email Service 개발자 안내서SES의 Amazon에서 이메일 주소 확인을 참조하세요.

- 자세한 API 내용은 명령 참조VerifyEmailIdentity의 섹션을 참조하세요. AWS CLI

# 를 사용하여 예제 보호 AWS CLI

다음 코드 예제에서는 Shield AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `associate-drt-log-bucket`

다음 코드 예시에서는 `associate-drt-log-bucket`을 사용하는 방법을 보여 줍니다.

AWS CLI

가 Amazon S3 버킷에 액세스할 DRT 수 있는 권한을 부여하려면

다음 `associate-drt-log-bucket` 예제에서는 DRT와 지정된 S3 버킷 간의 연결을 생성합니다. 이렇게 하면 DRT가 계정을 대신하여 버킷에 액세스할 수 있습니다.

```
aws shield associate-drt-log-bucket \
    --log-bucket flow-logs-for-website-lb
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Shield 고급 개발자 안내서의 DDoS 대응 팀 승인을 참조하세요.

- 자세한 API 내용은 명령 참조AssociateDrtLogBucket의 섹션을 참조하세요. AWS CLI

### `associate-drt-role`

다음 코드 예시에서는 `associate-drt-role`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자를 대신하여 잠재적 공격을 완화DRT할 수 있는 권한을 에 부여하려면

다음 `associate-drt-role` 예제에서는 DRT 와 지정된 역할 간의 연결을 생성합니다. 는 역할을 사용하여 계정에 액세스하고 관리할 DRT 수 있습니다.

```
aws shield associate-drt-role \
    --role-arn arn:aws:iam::123456789012:role/service-role/DrtRole
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Shield 고급 개발자 안내서의 DDoS 대응 팀 승인을 참조하세요.

- 자세한 API 내용은 명령 참조AssociateDrtRole의 섹션을 참조하세요. AWS CLI

### `create-protection`

다음 코드 예시에서는 `create-protection`을 사용하는 방법을 보여 줍니다.

AWS CLI

단일 AWS 리소스에 대해 AWS Shield Advanced 보호를 활성화하려면

다음 `create-protection` 예제에서는 지정된 AWS CloudFront 배포에 대해 Shield Advanced 보호를 활성화합니다.

```
aws shield create-protection \
    --name "Protection for CloudFront distribution" \
    --resource-arn arn:aws:cloudfront::123456789012:distribution/E198WC25FXOWY8
```

출력:

```
{
    "ProtectionId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

자세한 내용은 AWS Shield 고급 개발자 안내서의 보호할 리소스 지정을 참조하세요.

- 자세한 API 내용은 명령 참조CreateProtection의 섹션을 참조하세요. AWS CLI

### create-subscription

다음 코드 예시에서는 create-subscription을 사용하는 방법을 보여 줍니다.

AWS CLI

계정에 대해 AWS Shield Advanced 보호를 활성화하려면

다음 create-subscription 예제에서는 계정에 대해 Shield Advanced 보호를 활성화합니다.

```
aws shield create-subscription
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Shield Advanced 개발자 안내서의 Shield Advanced 시작하기를 참조하세요. AWS

- 자세한 API 내용은 명령 참조CreateSubscription의 섹션을 참조하세요. AWS CLI

### delete-protection

다음 코드 예시에서는 delete-protection을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 리소스에서 AWS Shield Advanced 보호를 제거하려면

다음 delete-protection 예제에서는 지정된 AWS Shield Advanced 보호를 제거합니다.

```
aws shield delete-protection \
    --protection-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Shield Advanced 개발자 안내서의 AWS 리소스에서 Shield Advanced 제거를 참조하세요. AWS

- 자세한 API 내용은 명령 참조DeleteProtection의 섹션을 참조하세요. AWS CLI

### describe-attack

다음 코드 예시에서는 describe-attack을 사용하는 방법을 보여 줍니다.

AWS CLI

공격에 대한 자세한 설명을 검색하려면

다음 describe-attack 예제에서는 지정된 DDoS 공격 ID를 사용하여 공격에 대한 세부 정보를 표시합니다. list-attacks 명령을 실행IDs하여 공격을 받을 수 있습니다.

```
aws shield describe-attack --attack-id a1b2c3d4-5678-90ab-cdef-EXAMPLE22222
```

출력:

```
{
    "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": []
    }
}
```

자세한 내용은 AWS Shield 고급 개발자 안내서의 DDoS 인시던트 검토를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeAttack의 섹션을 참조하세요. AWS CLI

### describe-drt-access

다음 코드 예시에서는 describe-drt-access을 사용하는 방법을 보여 줍니다.

AWS CLI

DRT가 사용자를 대신하여 공격을 완화하는 데 필요한 권한 부여에 대한 설명을 검색하려면

다음 describe-drt-access 예제는 가 DRT 보유한 역할 및 S3 버킷 권한을 검색하여 사용자를 대신하여 잠재적 공격에 대응할 수 있도록 합니다.

```
aws shield describe-drt-access
```

출력:

```
{
    "RoleArn": "arn:aws:iam::123456789012:role/service-role/DrtRole",
    "LogBucketList": [
        "flow-logs-for-website-lb"
    ]
}
```

자세한 내용은 AWS Shield 고급 개발자 안내서의 DDoS 대응 팀 승인을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeDrtAccess의 섹션을 참조하세요. AWS CLI

### describe-emergency-contact-settings

다음 코드 예시에서는 describe-emergency-contact-settings을 사용하는 방법을 보여 줍니다.

AWS CLI

에 저장된 긴급 이메일 주소를 검색하려면 DRT

다음 describe-emergency-contact-settings 예제에서는 DRT 계정의 에 저장된 이메일 주소를 검색합니다. 다음은 의심되는 공격에 대응할 때 가 DRT 연락해야 하는 주소입니다.

```
aws shield describe-emergency-contact-settings
```

출력:

```
{
    "EmergencyContactList": [
        {
            "EmailAddress": "ops@example.com"
        },
        {
            "EmailAddress": "ddos-notifications@example.com"
       }
    ]
}
```

자세한 내용은 AWS Shield AWS 고급 개발자 안내서의 Shield 작동 방식<https://docs.aws.amazon.com/waf/latest/developerguide/ddos-overview.html>을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeEmergencyContactSettings의 섹션을 참조하세요. AWS CLI

### describe-protection

다음 코드 예시에서는 describe-protection을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS Shield Advanced 보호에 대한 세부 정보를 검색하려면

다음 describe-protection 예제에서는 지정된 ID로 Shield Advanced 보호에 대한 세부 정보를 표시합니다. list-protections 명령을 실행IDs하여 보호를 받을 수 있습니다.

```
aws shield describe-protection \
    --protection-id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

출력:

```
{
    "Protection": {
        "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "Name": "1.2.3.4",
        "ResourceArn": "arn:aws:ec2:us-west-2:123456789012:eip-allocation/
eipalloc-0ac1537af40742a6d"
    }
}
```

자세한 내용은 AWS Shield 고급 개발자 안내서의 보호할 리소스 지정을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeProtection의 섹션을 참조하세요. AWS CLI

### describe-subscription

다음 코드 예시에서는 describe-subscription을 사용하는 방법을 보여 줍니다.

AWS CLI

계정에 대한 AWS Shield Advanced 보호의 세부 정보를 검색하려면

다음 describe-subscription 예제에서는 계정에 제공된 Shield Advanced 보호에 대한 세부 정보를 보여줍니다.

```
aws shield describe-subscription
```

출력:

```
{
    "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
            }
        ]
    }
}
```

자세한 내용은 AWS Shield 고급 개발자 안내서의 Shield 작동 방식을 참조하세요. AWS

- 자세한 API 내용은 명령 참조DescribeSubscription의 섹션을 참조하세요. AWS CLI

### `disassociate-drt-log-bucket`

다음 코드 예시에서는 `disassociate-drt-log-bucket`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자를 대신하여 Amazon S3 버킷에 액세스할 DRT 수 있는 권한을 제거하려면

다음 `disassociate-drt-log-bucket` 예제에서는 DRT와 지정된 S3 버킷 간의 연결을 제거합니다. 이 명령이 완료되면 는 더 이상 계정을 대신하여 버킷에 액세스할 DRT 수 없습니다.

```
aws shield disassociate-drt-log-bucket \
    --log-bucket flow-logs-for-website-lb
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Shield 고급 개발자 안내서의 DDoS 대응 팀 승인을 참조하세요.

- 자세한 API 내용은 명령 참조DisassociateDrtLogBucket의 섹션을 참조하세요. AWS CLI

### `disassociate-drt-role`

다음 코드 예시에서는 `disassociate-drt-role`을 사용하는 방법을 보여 줍니다.

AWS CLI

에 대한 권한 부여를 제거하여 사용자를 대신하여 잠재적 공격을 완화DRT하려면

다음 `disassociate-drt-role` 예제에서는 DRT 와 계정 간의 연결을 제거합니다. 이 호출 후에는 DRT가 더 이상 계정에 액세스하거나 계정을 관리할 수 없습니다.

```
aws shield disassociate-drt-role
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Shield 고급 개발자 안내서의 DDoS 대응 팀 승인을 참조하세요.

- 자세한 API 내용은 명령 참조DisassociateDrtRole의 섹션을 참조하세요. AWS CLI

### `get-subscription-state`

다음 코드 예시에서는 `get-subscription-state`을 사용하는 방법을 보여 줍니다.

AWS CLI

계정의 AWS Shield Advanced 구독의 현재 상태를 검색하려면

다음 `get-subscription-state` 예제에서는 계정에 대한 Shield Advanced 보호 상태를 검색합니다.

```
aws shield get-subscription-state
```

출력:

```
{
    "SubscriptionState": "ACTIVE"
```

```
}
```

자세한 내용은 AWS Shield 고급 개발자 안내서의 Shield 작동 방식을 참조하세요. AWS

- 자세한 API 내용은 명령 참조GetSubscriptionState의 섹션을 참조하세요. AWS CLI

### `list-attacks`

다음 코드 예시에서는 `list-attacks`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS Shield Advanced에서 공격 요약을 검색하려면

다음 `list-attacks` 예제에서는 지정된 기간 동안 지정된 AWS CloudFront 배포에 대한 공격 요약을 검색합니다. 응답에는 공격에 대한 자세한 정보를 위해 `describe-attack` 명령에 제공할 수 IDs 있는 공격이 포함됩니다.

```
aws shield list-attacks \
    --resource-arns arn:aws:cloudfront::12345678910:distribution/E1PXMP22ZVFAOR \
    --start-time FromInclusive=1529280000,ToExclusive=1529300000
```

출력:

```
{
    "AttackSummaries": [
        {
            "AttackId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "ResourceArn": "arn:aws:cloudfront::123456789012:distribution/
E1PXMP22ZVFAOR",
            "StartTime": 1529280000.0,
            "EndTime": 1529449200.0,
            "AttackVectors": [
                {
                    "VectorType": "SYN_FLOOD"
                }
            ]
        }
    ]
}
```

자세한 내용은 AWS Shield 고급 개발자 안내서의 DDoS 인시던트 검토를 참조하세요.

- 자세한 API 내용은 명령 참조ListAttacks의 섹션을 참조하세요. AWS CLI

### `list-protections`

다음 코드 예시에서는 `list-protections`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS Shield Advanced에서 보호 요약을 검색하려면

다음 `list-protections` 예제에서는 계정에 대해 활성화된 보호에 대한 요약을 검색합니다.

```
aws shield list-protections
```

출력:

```
{
    "Protections": [
        {
            "Id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
            "Name": "Protection for CloudFront distribution",
            "ResourceArn": "arn:aws:cloudfront::123456789012:distribution/
E198WC25FXOWY8"
        }
    ]
}
```

자세한 내용은 AWS Shield 고급 개발자 안내서의 보호할 리소스 지정을 참조하세요.

- 자세한 API 내용은 명령 참조ListProtections의 섹션을 참조하세요. AWS CLI

### `update-emergency-contact-settings`

다음 코드 예시에서는 `update-emergency-contact-settings`을 사용하는 방법을 보여 줍니다.

AWS CLI

에 저장된 긴급 이메일 주소를 정의하려면 DRT

다음 `update-emergency-contact-settings` 예제에서는 의심되는 공격에 대응할 때 DRT가 연락해야 하는 두 개의 이메일 주소를 정의합니다.

```
aws shield update-emergency-contact-settings \
        --emergency-contact-list EmailAddress=ops@example.com EmailAddress=ddos-
notifications@example.com
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Shield 고급 개발자 안내서의 Shield 작동 방식을 참조하세요. AWS

- 자세한 API 내용은 명령 참조UpdateEmergencyContactSettings의 섹션을 참조하세요. AWS CLI

### update-subscription

다음 코드 예시에서는 update-subscription을 사용하는 방법을 보여 줍니다.

AWS CLI

계정의 AWS Shield Advanced 구독을 수정하려면

다음 update-subscription 예제에서는 계정에 대한 AWS Shield Advanced 구독의 자동 갱신을 활성화합니다.

```
aws shield update-subscription \
    --auto-renew ENABLED
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Shield 고급 개발자 안내서의 Shield 작동 방식을 참조하세요. AWS

- 자세한 API 내용은 명령 참조UpdateSubscription의 섹션을 참조하세요. AWS CLI

## 를 사용한 서명자 예제 AWS CLI

다음 코드 예제에서는 서명자와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `cancel-signing-profile`

다음 코드 예시에서는 `cancel-signing-profile`을 사용하는 방법을 보여 줍니다.

AWS CLI

서명 프로파일을 삭제하려면

다음 `cancel-signing-profile` 예제에서는 AWS 서명자에서 기존 서명 프로파일을 제거합니다.

```
aws signer cancel-signing-profile \
    --profile-name MyProfile1
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조CancelSigningProfile의 섹션을 참조하세요. AWS CLI

### `describe-signing-job`

다음 코드 예시에서는 `describe-signing-job`을 사용하는 방법을 보여 줍니다.

AWS CLI

서명 작업에 대한 세부 정보를 표시하려면

다음 `describe-signing-job` 예제에서는 지정된 서명 작업에 대한 세부 정보를 표시합니다.

```
aws signer describe-signing-job \
    --job-id 2065c468-73e2-4385-a6c9-0123456789abc
```

출력:

```
{
    "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": "PNyFaUTgsQh5ZdMCcoCe6pT1gOpgB_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
}
```

- 자세한 API 내용은 명령 참조DescribeSigningJob의 섹션을 참조하세요. AWS CLI

### `get-signing-platform`

다음 코드 예시에서는 `get-signing-platform`을 사용하는 방법을 보여 줍니다.

AWS CLI

서명 플랫폼에 대한 세부 정보를 표시하려면

다음 `get-signing-platform` 예제에서는 지정된 서명 플랫폼에 대한 세부 정보를 표시합니다.

```
aws signer get-signing-platform \
    --platform-id AmazonFreeRTOS-TI-CC3220SF
```

출력:

```
{
    "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"
        ]
    }
}
```

- 자세한 API 내용은 명령 참조GetSigningPlatform의 섹션을 참조하세요. AWS CLI

### get-signing-profile

다음 코드 예시에서는 get-signing-profile을 사용하는 방법을 보여 줍니다.

AWS CLI

서명 프로파일에 대한 세부 정보를 표시하려면

다음 get-signing-profile 예제에서는 지정된 서명 프로파일에 대한 세부 정보를 표시합니다.

```
aws signer get-signing-profile \
    --profile-name MyProfile3
```

출력:

```
{
    "platformId": "AmazonFreeRTOS-TI-CC3220SF",
    "profileName": "MyProfile3",
    "status": "Active",
    "signingMaterial": {
        "certificateArn": "arn:aws:acm:us-
west-2:123456789012:certificate/6a55389b-306b-4e8c-a95c-0123456789abc"
    }
}
```

- 자세한 API 내용은 명령 참조GetSigningProfile의 섹션을 참조하세요. AWS CLI

### list-signing-jobs

다음 코드 예시에서는 list-signing-jobs을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 서명 작업을 나열하려면

다음 list-signing-jobs 예제에서는 계정의 모든 서명 작업에 대한 세부 정보를 표시합니다.

```
aws signer list-signing-jobs
```

이 예제에서는 두 개의 작업이 반환됩니다. 하나는 성공이고 다른 하나는 실패입니다.

```
{
    "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": "PNyFaUTgsQh5ZdMCcoCe6pT1gOpgB_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": "PNyFaUTgsQh5ZdMCcoCe6pT1gOpgB_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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListSigningJobs의 섹션을 참조하세요. AWS CLI

### `list-signing-platforms`

다음 코드 예시에서는 `list-signing-platforms`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 서명 플랫폼을 나열하려면

다음 `list-signing-platforms` 예제에서는 사용 가능한 모든 서명 플랫폼에 대한 세부 정보를 표시합니다.

```
aws signer list-signing-platforms
```

출력:

```
{
    "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"
                ]
            }
        }
```

```
    ]
}
```

- 자세한 API 내용은 명령 참조ListSigningPlatforms의 섹션을 참조하세요. AWS CLI

### `list-signing-profiles`

다음 코드 예시에서는 `list-signing-profiles`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 서명 프로파일을 나열하려면

다음 `list-signing-profiles` 예제에서는 계정의 모든 서명 프로파일에 대한 세부 정보를 표시합니다.

```
aws signer list-signing-profiles
```

출력:

```
{
    "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"
            }
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListSigningProfiles의 섹션을 참조하세요. AWS CLI

### put-signing-profile

다음 코드 예시에서는 put-signing-profile을 사용하는 방법을 보여 줍니다.

AWS CLI

서명 프로파일을 생성하려면

다음 put-signing-profile 예제에서는 지정된 인증서 및 플랫폼을 사용하여 서명 프로파일을 생성합니다.

```
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
```

출력:

```
{
    "arn": "arn:aws:signer:us-west-2:123456789012:/signing-profiles/MyProfile6"
}
```

- 자세한 API 내용은 명령 참조PutSigningProfile의 섹션을 참조하세요. AWS CLI

### start-signing-job

다음 코드 예시에서는 start-signing-job을 사용하는 방법을 보여 줍니다.

AWS CLI

서명 작업을 시작하려면

다음 start-signing-job 예제에서는 지정된 소스에서 찾은 코드에서 서명 작업을 시작합니다. 지정된 프로파일을 사용하여 서명을 수행하고 서명된 코드를 지정된 대상에 배치합니다.

```
aws signer start-signing-job \
    --source 's3={bucketName=signer-
source,key=MyCode.rb,version=PNyFaUTgsQh5ZdMCcoCe6pT1gOpgB_M4}' \
```

```
    --destination 's3={bucketName=signer-destination,prefix=signed-}' \
    --profile-name MyProfile7
```

출력은 서명 작업의 ID입니다.

```
{
    "jobId": "2065c468-73e2-4385-a6c9-0123456789abc"
}
```

- 자세한 API 내용은 명령 참조StartSigningJob의 섹션을 참조하세요. AWS CLI

## 를 사용한 Snowball 예제 AWS CLI

다음 코드 예제에서는 Snowball과 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `get-snowball-usage`

다음 코드 예시에서는 `get-snowball-usage`을 사용하는 방법을 보여 줍니다.

AWS CLI

계정의 Snowball 서비스 제한에 대한 정보를 가져오려면

다음 `get-snowball-usage` 예제에서는 계정의 Snowball 서비스 제한과 계정이 사용 중인 Snowball 수에 대한 정보를 표시합니다.

```
aws snowball get-snowball-usage
```

출력:

```
{
    "SnowballLimit": 1,
    "SnowballsInUse": 0
}
```

FOR 자세한 내용은 AWS Snowball 개발자 안내서의 Snowball 엣지 제한을 참조하세요. AWS

- 자세한 API 내용은 명령 참조GetSnowballUsage의 섹션을 참조하세요. AWS CLI

### `list-jobs`

다음 코드 예시에서는 `list-jobs`을 사용하는 방법을 보여 줍니다.

AWS CLI

계정의 현재 Snowball 작업을 나열하려면

다음 `list-jobs` 예제에서는 `JobListEntry` 객체 배열을 보여줍니다. 이 예제에서는 단일 작업이 나열됩니다.

```
aws snowball list-jobs
```

출력:

```
{
    "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"
        }
   ]
}
```

자세한 내용은 AWS Snowball 개발자 안내서의 Snowball Edge 디바이스용 작업을 참조하세요. AWS

- 자세한 API 내용은 명령 참조ListJobs의 섹션을 참조하세요. AWS CLI

## 를 사용한 Amazon SNS 예제 AWS CLI

다음 코드 예제에서는 Amazon 와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다SNS.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

시나리오는 동일한 서비스 내에서 또는 다른 AWS 서비스와 결합된 상태에서 여러 함수를 호출하여 특정 태스크를 수행하는 방법을 보여주는 코드 예제입니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업
- 시나리오

### 작업

#### `add-permission`

다음 코드 예시에서는 `add-permission`을 사용하는 방법을 보여 줍니다.

AWS CLI

주제에 권한을 추가하려면

다음 `add-permission` 예제에서는 AWS 계정 에서 지정된 주제로 `Publish` 작업을 사용할 987654321098 수 있는 권한을 AWS 계정에 추가합니다123456789012.

```
aws sns add-permission \
    --topic-arn arn:aws:sns:us-west-2:123456789012:MyTopic \
    --label Publish-Permission \
    --aws-account-id 987654321098 \
    --action-name Publish
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조AddPermission의 섹션을 참조하세요. AWS CLI

### check-if-phone-number-is-opted-out

다음 코드 예시에서는 check-if-phone-number-is-opted-out을 사용하는 방법을 보여 줍니다.

AWS CLI

전화번호에 대한 SMS 메시지 옵트아웃을 확인하려면

다음 check-if-phone-number-is-opted-out 예제에서는 지정된 전화번호가 현재 AWS 계정에서 SMS 메시지 수신을 거부하는지 확인합니다.

```
aws sns check-if-phone-number-is-opted-out \
    --phone-number +1555550100
```

출력:

```
{
    "isOptedOut": false
}
```

- 자세한 API 내용은 명령 참조CheckIfPhoneNumberIsOptedOut의 섹션을 참조하세요. AWS CLI

### confirm-subscription

다음 코드 예시에서는 confirm-subscription을 사용하는 방법을 보여 줍니다.

AWS CLI

구독을 확인하려면

다음 confirm-subscription 명령은 라는 SNS 주제를 구독할 때 시작된 확인 프로세스를 완료합니다my-topic. --token 파라미터는 구독 호출에 지정된 알림 엔드포인트로 전송된 확인 메시지에서 제공됩니다.

```
aws sns confirm-subscription \
    --topic-arn arn:aws:sns:us-west-2:123456789012:my-topic \
```

```
    --
token 2336412f37fb687f5d51e6e241d7700ae02f7124d8268910b858cb4db727ceeb2474bb937929d3bdd7ce5d
```

출력:

```
{
    "SubscriptionArn": "arn:aws:sns:us-west-2:123456789012:my-
topic:8a21d249-4329-4871-acc6-7be709c6ea7f"
}
```

- 자세한 API 내용은 명령 참조ConfirmSubscription의 섹션을 참조하세요. AWS CLI

### create-platform-application

다음 코드 예시에서는 create-platform-application을 사용하는 방법을 보여 줍니다.

AWS CLI

플랫폼 애플리케이션을 생성하려면

다음 create-platform-application 예제에서는 지정된 플랫폼 자격 증명을 사용하여 Google Firebase 플랫폼 애플리케이션을 생성합니다.

```
aws sns create-platform-application \
    --name MyApplication \
    --platform GCM \
    --attributes PlatformCredential=EXAMPLEabcd12345jklm67890stuv12345bcdef
```

출력:

```
{
    "PlatformApplicationArn": "arn:aws:sns:us-west-2:123456789012:app/GCM/
MyApplication"
}
```

- 자세한 API 내용은 명령 참조CreatePlatformApplication의 섹션을 참조하세요. AWS CLI

### create-topic

다음 코드 예시에서는 create-topic을 사용하는 방법을 보여 줍니다.

AWS CLI

SNS 주제를 생성하려면

다음 `create-topic` 예제에서는 이라는 SNS 주제를 생성합니다`my-topic`.

```
aws sns create-topic \
    --name my-topic
```

출력:

```
{
    "ResponseMetadata": {
        "RequestId": "1469e8d7-1642-564e-b85d-a19b4b341f83"
    },
    "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-topic"
}
```

자세한 내용은 AWS 명령줄 인터페이스 사용 설명서의 Amazon SQS 및 Amazon에서 SNS 명령줄 인터페이스 사용을 참조하세요. AWS

- 자세한 API 내용은 명령 참조CreateTopic의 섹션을 참조하세요. AWS CLI

### `delete-endpoint`

다음 코드 예시에서는 `delete-endpoint`을 사용하는 방법을 보여 줍니다.

AWS CLI

플랫폼 애플리케이션 엔드포인트를 삭제하려면

다음 `delete-endpoint` 예제에서는 지정된 플랫폼 애플리케이션 엔드포인트를 삭제합니다.

```
aws sns delete-endpoint \
    --endpoint-arn arn:aws:sns:us-west-2:123456789012:endpoint/GCM/
MyApplication/12345678-abcd-9012-efgh-345678901234
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeleteEndpoint의 섹션을 참조하세요. AWS CLI

### delete-platform-application

다음 코드 예시에서는 delete-platform-application을 사용하는 방법을 보여 줍니다.

AWS CLI

플랫폼 애플리케이션을 삭제하려면

다음 delete-platform-application 예제에서는 지정된 플랫폼 애플리케이션을 삭제합니다.

```
aws sns delete-platform-application \
    --platform-application-arn arn:aws:sns:us-west-2:123456789012:app/ADM/
MyApplication
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeletePlatformApplication의 섹션을 참조하세요. AWS CLI

### delete-topic

다음 코드 예시에서는 delete-topic을 사용하는 방법을 보여 줍니다.

AWS CLI

SNS 주제를 삭제하려면

다음 delete-topic 예제에서는 지정된 SNS 주제를 삭제합니다.

```
aws sns delete-topic \
    --topic-arn "arn:aws:sns:us-west-2:123456789012:my-topic"
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeleteTopic의 섹션을 참조하세요. AWS CLI

### get-endpoint-attributes

다음 코드 예시에서는 get-endpoint-attributes을 사용하는 방법을 보여 줍니다.

AWS CLI

플랫폼 애플리케이션 엔드포인트 속성을 나열하려면

다음 get-endpoint-attributes 예제에서는 지정된 플랫폼 애플리케이션 엔드포인트의 속성을 나열합니다.

```
aws sns get-endpoint-attributes \
    --endpoint-arn arn:aws:sns:us-west-2:123456789012:endpoint/GCM/
MyApplication/12345678-abcd-9012-efgh-345678901234
```

출력:

```
{
    "Attributes": {
        "Enabled": "true",
        "Token": "EXAMPLE12345..."
    }
}
```

- 자세한 API 내용은 명령 참조GetEndpointAttributes의 섹션을 참조하세요. AWS CLI

### get-platform-application-attributes

다음 코드 예시에서는 get-platform-application-attributes을 사용하는 방법을 보여 줍니다.

AWS CLI

플랫폼 애플리케이션 속성을 나열하려면

다음 get-platform-application-attributes 예제에서는 지정된 플랫폼 애플리케이션의 속성을 나열합니다.

```
aws sns get-platform-application-attributes \
    --platform-application-arn arn:aws:sns:us-west-2:123456789012:app/MPNS/
MyApplication
```

출력:

```
{
    "Attributes": {
        "Enabled": "true",
        "SuccessFeedbackSampleRate": "100"
    }
```

```
}
```

- 자세한 API 내용은 명령 참조GetPlatformApplicationAttributes의 섹션을 참조하세요. AWS CLI

### get-sms-attributes

다음 코드 예시에서는 get-sms-attributes을 사용하는 방법을 보여 줍니다.

AWS CLI

기본 SMS 메시지 속성을 나열하려면

다음 get-sms-attributes 예제에서는 SMS 메시지 전송을 위한 기본 속성을 나열합니다.

```
aws sns get-sms-attributes
```

출력:

```
{
    "attributes": {
        "DefaultSenderID": "MyName"
    }
}
```

- 자세한 API 내용은 AWS CLI 명령 참조 의 GetSMSAttributes를 참조하세요.

### get-subscription-attributes

다음 코드 예시에서는 get-subscription-attributes을 사용하는 방법을 보여 줍니다.

AWS CLI

주제에 대한 구독 속성을 검색하려면

다음은 지정된 구독의 속성을 get-subscription-attributes 표시합니다. list-subscriptions 명령의 출력subscription-arn에서 를 가져올 수 있습니다.

```
aws sns get-subscription-attributes \
    --subscription-arn "arn:aws:sns:us-west-2:123456789012:my-
topic:8a21d249-4329-4871-acc6-7be709c6ea7f"
```

출력:

```
{
    "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"
    }
}
```

- 자세한 API 내용은 명령 참조GetSubscriptionAttributes의 섹션을 참조하세요. AWS CLI

### get-topic-attributes

다음 코드 예시에서는 get-topic-attributes을 사용하는 방법을 보여 줍니다.

AWS CLI

주제의 속성을 검색하려면

다음 get-topic-attributes 예제에서는 지정된 주제의 속성을 표시합니다.

```
aws sns get-topic-attributes \
    --topic-arn "arn:aws:sns:us-west-2:123456789012:my-topic"
```

출력:

```
{
    "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"
    }
}
```

- 자세한 API 내용은 명령 참조GetTopicAttributes의 섹션을 참조하세요. AWS CLI

### `list-endpoints-by-platform-application`

다음 코드 예시에서는 `list-endpoints-by-platform-application`을 사용하는 방법을 보여 줍니다.

AWS CLI

플랫폼 애플리케이션의 엔드포인트를 나열하려면

다음 `list-endpoints-by-platform-application` 예제에서는 지정된 플랫폼 애플리케이션의 엔드포인트 및 엔드포인트 속성을 나열합니다.

```
aws sns list-endpoints-by-platform-application \
    --platform-application-arn arn:aws:sns:us-west-2:123456789012:app/GCM/
MyApplication
```

출력:

```
{
    "Endpoints": [
        {
            "Attributes": {
                "Token": "EXAMPLE12345...,
                "Enabled": "true"
            },
            "EndpointArn": "arn:aws:sns:us-west-2:123456789012:endpoint/GCM/
MyApplication/12345678-abcd-9012-efgh-345678901234"
```

```
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListEndpointsByPlatformApplication의 섹션을 참조하세요. AWS CLI

### `list-phone-numbers-opted-out`

다음 코드 예시에서는 `list-phone-numbers-opted-out`을 사용하는 방법을 보여 줍니다.

AWS CLI

SMS 메시지 옵트아웃을 나열하려면

다음 `list-phone-numbers-opted-out` 예제에서는 SMS 메시지 수신을 거부한 전화번호를 나열합니다.

```
aws sns list-phone-numbers-opted-out
```

출력:

```
{
    "phoneNumbers": [
        "+15555550100"
    ]
}
```

- 자세한 API 내용은 명령 참조ListPhoneNumbersOptedOut의 섹션을 참조하세요. AWS CLI

### `list-platform-applications`

다음 코드 예시에서는 `list-platform-applications`을 사용하는 방법을 보여 줍니다.

AWS CLI

플랫폼 애플리케이션을 나열하려면

다음 `list-platform-applications` 예제에서는 ADM 및 의 플랫폼 애플리케이션을 나열합니다MPNS.

```
aws sns list-platform-applications
```

출력:

```
{
    "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"
            }
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListPlatformApplications의 섹션을 참조하세요. AWS CLI

### `list-subscriptions-by-topic`

다음 코드 예시에서는 `list-subscriptions-by-topic`을 사용하는 방법을 보여 줍니다.

AWS CLI

주제와 연결된 구독을 나열하려면

다음은 지정된 주제와 연결된 SNS 구독 목록을 `list-subscriptions-by-topic` 검색합니다.

```
aws sns list-subscriptions-by-topic \
    --topic-arn "arn:aws:sns:us-west-2:123456789012:my-topic"
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListSubscriptionsByTopic의 섹션을 참조하세요. AWS CLI

### `list-subscriptions`

다음 코드 예시에서는 `list-subscriptions`을 사용하는 방법을 보여 줍니다.

AWS CLI

SNS 구독을 나열하려면

다음 `list-subscriptions` 예제에서는 AWS 계정의 SNS 구독 목록을 표시합니다.

```
aws sns list-subscriptions
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListSubscriptions의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

주제에 대한 태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 지정된 Amazon SNS 주제에 대한 태그를 나열합니다.

```
aws sns list-tags-for-resource \
    --resource-arn arn:aws:sns:us-west-2:123456789012:MyTopic
```

출력:

```
{
    "Tags": [
        {
            "Key": "Team",
            "Value": "Alpha"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### `list-topics`

다음 코드 예시에서는 `list-topics`을 사용하는 방법을 보여 줍니다.

AWS CLI

SNS 주제를 나열하려면

다음 `list-topics` 예제에서는 AWS 계정의 모든 SNS 주제를 나열합니다.

```
aws sns list-topics
```

출력:

```
{
    "Topics": [
        {
            "TopicArn": "arn:aws:sns:us-west-2:123456789012:my-topic"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListTopics의 섹션을 참조하세요. AWS CLI

### opt-in-phone-number

다음 코드 예시에서는 opt-in-phone-number을 사용하는 방법을 보여 줍니다.

AWS CLI

SMS 메시지 수신을 옵트인하려면

다음 opt-in-phone-number 예제에서는 지정된 전화번호를 SMS 수신 메시지로 선택합니다.

```
aws sns opt-in-phone-number \
    --phone-number +15555550100
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조OptInPhoneNumber의 섹션을 참조하세요. AWS CLI

### publish

다음 코드 예시에서는 publish을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 주제에 메시지를 게시하려면

다음 publish 예제에서는 지정된 메시지를 지정된 SNS 주제에 게시합니다. 메시지는 줄 바꿈을 포함할 수 있는 텍스트 파일에서 제공됩니다.

```
aws sns publish \
```

```
    --topic-arn "arn:aws:sns:us-west-2:123456789012:my-topic" \
    --message file://message.txt
```

message.txt의 콘텐츠:

```
Hello World
Second Line
```

출력:

```
{
    "MessageId": "123a45b6-7890-12c3-45d6-111122223333"
}
```

예제 2: 전화번호에 SMS 메시지를 게시하려면

다음 publish 예제에서는 Hello world! 메시지를 전화번호 +1-555-555-0100에 게시합니다.

```
aws sns publish \
    --message "Hello world!" \
    --phone-number +1-555-555-0100
```

출력:

```
{
    "MessageId": "123a45b6-7890-12c3-45d6-333322221111"
}
```

- 자세한 API 내용은 AWS CLI 명령 참조의 게시를 참조하세요.

### put-data-protection-policy

다음 코드 예시에서는 put-data-protection-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

데이터 보호 정책을 설정하려면

예제 1: 게시자가 를 사용하여 메시지를 게시하지 못하도록 거부하려면 CreditCardNumber

다음 put-data-protection-policy 예제에서는 게시자가 를 사용하여 메시지를 게시하는 것을 거부합니다 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\":{}}}]}"
```

이 명령은 출력을 생성하지 않습니다.

예제 2: 파일에서 파라미터를 로드하는 방법

다음은 파일에서 파라미터를 put-data-protection-policy 로드합니다.

```
aws sns put-data-protection-policy \
    --resource-arn arn:aws:sns:us-west-2:123456789012:MyTopic \
    --data-protection-policy file://policy.json
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조PutDataProtectionPolicy의 섹션을 참조하세요. AWS CLI

### remove-permission

다음 코드 예시에서는 remove-permission을 사용하는 방법을 보여 줍니다.

AWS CLI

주제에서 권한을 제거하려면

다음 remove-permission 예제에서는 지정된 주제Publish-Permission에서 권한을 제거합니다.

```
aws sns remove-permission \
    --topic-arn arn:aws:sns:us-west-2:123456789012:MyTopic \
    --label Publish-Permission
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조RemovePermission의 섹션을 참조하세요. AWS CLI

### set-endpoint-attributes

다음 코드 예시에서는 set-endpoint-attributes을 사용하는 방법을 보여 줍니다.

AWS CLI

엔드포인트 속성을 설정하려면

다음 set-endpoint-attributes 예제에서는 지정된 플랫폼 애플리케이션 엔드포인트를 비활성화합니다.

```
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
```

출력:

```
{
    "Attributes": {
        "Enabled": "false",
        "Token": "EXAMPLE12345..."
    }
}
```

- 자세한 API 내용은 명령 참조SetEndpointAttributes의 섹션을 참조하세요. AWS CLI

### set-platform-application-attributes

다음 코드 예시에서는 set-platform-application-attributes을 사용하는 방법을 보여 줍니다.

AWS CLI

플랫폼 애플리케이션 속성을 설정하려면

다음 set-platform-application-attributes 예제에서는 지정된 플랫폼 애플리케이션의 EventDeliveryFailure 속성을 지정된 Amazon SNS 주제ARN의 로 설정합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조SetPlatformApplicationAttributes의 섹션을 참조하세요. AWS CLI

### set-sms-attributes

다음 코드 예시에서는 set-sms-attributes을 사용하는 방법을 보여 줍니다.

AWS CLI

SMS 메시지 속성을 설정하려면

다음 set-sms-attributes 예제에서는 SMS 메시지의 기본 발신자 ID를 로 설정합니다MyName.

```
aws sns set-sms-attributes \
    --attributes DefaultSenderID=MyName
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 AWS CLI 명령 참조의 SetSMSAttributes를 참조하세요.

### set-subscription-attributes

다음 코드 예시에서는 set-subscription-attributes을 사용하는 방법을 보여 줍니다.

AWS CLI

구독 속성을 설정하려면

다음 set-subscription-attributes 예제에서는 RawMessageDelivery 속성을 SQS 구독으로 설정합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

다음 set-subscription-attributes 예제에서는 FilterPolicy 속성을 SQS 구독으로 설정합니다.

```
aws sns set-subscription-attributes \
    --subscription-arn arn:aws:sns:us-
east-1:123456789012:mytopic:f248de18-2cf6-578c-8592-b6f1eaa877dc \
    --attribute-name FilterPolicy \
    --attribute-value "{ \"anyMandatoryKey\": [\"any\", \"of\", \"these\"] }"
```

이 명령은 출력을 생성하지 않습니다.

다음 set-subscription-attributes 예제에서는 SQS 구독에서 FilterPolicy 속성을 제거합니다.

```
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 "{}"
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조SetSubscriptionAttributes의 섹션을 참조하세요. AWS CLI

### set-topic-attributes

다음 코드 예시에서는 set-topic-attributes을 사용하는 방법을 보여 줍니다.

AWS CLI

주제에 대한 속성을 설정하려면

다음 set-topic-attributes 예제에서는 지정된 주제에 DisplayName 속성을 설정합니다.

```
aws sns set-topic-attributes \
    --topic-arn arn:aws:sns:us-west-2:123456789012:MyTopic \
    --attribute-name DisplayName \
    --attribute-value MyTopicDisplayName
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조SetTopicAttributes의 섹션을 참조하세요. AWS CLI

### subscribe

다음 코드 예시에서는 subscribe을 사용하는 방법을 보여 줍니다.

AWS CLI

주제를 구독하려면

다음 subscribe 명령은 이메일 주소로 지정된 주제를 구독합니다.

```
aws sns subscribe \
    --topic-arn arn:aws:sns:us-west-2:123456789012:my-topic \
    --protocol email \
    --notification-endpoint my-email@example.com
```

출력:

```
{
    "SubscriptionArn": "pending confirmation"
}
```

- API 자세한 내용은 AWS CLI 명령 참조의 구독을 참조하세요.

### tag-resource

다음 코드 예시에서는 tag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

주제에 태그를 추가하려면

다음 tag-resource 예제에서는 지정된 Amazon SNS 주제에 메타데이터 태그를 추가합니다.

```
aws sns tag-resource \
    --resource-arn arn:aws:sns:us-west-2:123456789012:MyTopic \
    --tags Key=Team,Value=Alpha
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### unsubscribe

다음 코드 예시에서는 unsubscribe을 사용하는 방법을 보여 줍니다.

AWS CLI

주제 구독을 취소하려면

다음 unsubscribe 예제에서는 주제에서 지정된 구독을 삭제합니다.

```
aws sns unsubscribe \
    --subscription-arn arn:aws:sns:us-west-2:0123456789012:my-
topic:8a21d249-4329-4871-acc6-7be709c6ea7f
```

이 명령은 출력을 생성하지 않습니다.

- API 자세한 내용은 AWS CLI 명령 참조의 구독 취소를 참조하세요.

### untag-resource

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

주제에서 태그를 제거하려면

다음 untag-resource 예제에서는 지정된 Amazon SNS 주제에서 지정된 키가 있는 모든 태그를 제거합니다.

```
aws sns untag-resource \
    --resource-arn arn:aws:sns:us-west-2:123456789012:MyTopic \
    --tag-keys Team
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

## 시나리오

푸시 알림에 대한 플랫폼 엔드포인트 생성

다음 코드 예제는 Amazon SNS 푸시 알림을 위한 플랫폼 엔드포인트를 생성하는 방법을 보여줍니다.

AWS CLI

플랫폼 애플리케이션 엔드포인트를 생성하려면

다음 `create-platform-endpoint` 예제에서는 지정된 토큰을 사용하여 지정된 플랫폼 애플리케이션의 엔드포인트를 생성합니다.

```
aws sns create-platform-endpoint \
    --platform-application-arn arn:aws:sns:us-west-2:123456789012:app/GCM/
MyApplication \
    --token EXAMPLE12345...
```

출력:

```
{
      "EndpointArn": "arn:aws:sns:us-west-2:1234567890:endpoint/GCM/
MyApplication/12345678-abcd-9012-efgh-345678901234"
}
```

# 를 사용한 Amazon SQS 예제 AWS CLI

다음 코드 예제에서는 Amazon 와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다SQS.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### `add-permission`

다음 코드 예시에서는 `add-permission`을 사용하는 방법을 보여 줍니다.

AWS CLI

대기열에 권한을 추가하려면

이 예제에서는 지정된 AWS 계정이 지정된 대기열로 메시지를 보낼 수 있도록 합니다.

명령:

```
aws sqs add-permission --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/
MyQueue --label SendMessagesFromMyQueue --aws-account-ids 12345EXAMPLE --
actions SendMessage
```

출력:

```
None.
```

- 자세한 API 내용은 명령 참조AddPermission의 섹션을 참조하세요. AWS CLI

### cancel-message-move-task

다음 코드 예시에서는 cancel-message-move-task을 사용하는 방법을 보여 줍니다.

AWS CLI

메시지 이동 작업을 취소하려면

다음 cancel-message-move-task 예제에서는 지정된 메시지 이동 작업을 취소합니다.

```
aws sqs cancel-message-move-task \
    --task-handle AQEB6nR4...HzlvZQ==
```

출력:

```
{
    "ApproximateNumberOfMessagesMoved": 102
}
```

자세한 내용은 개발자 안내서의 Amazon SQS API 권한: 작업 및 리소스 참조를 참조하세요.

- 자세한 API 내용은 명령 참조CancelMessageMoveTask의 섹션을 참조하세요. AWS CLI

### change-message-visibility-batch

다음 코드 예시에서는 change-message-visibility-batch을 사용하는 방법을 보여 줍니다.

AWS CLI

여러 메시지의 제한 시간 가시성을 배치로 변경하려면

이 예제에서는 2개의 지정된 메시지의 제한 시간 가시성을 10시간(10시간*60분*60초)으로 변경합니다.

명령:

```
aws sqs change-message-visibility-batch --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --entries file://change-message-visibility-batch.json
```

입력 파일(change-message-visibility-batch.json):

```
[
  {
    "Id": "FirstMessage",
        "ReceiptHandle": "AQEBhz2q...Jf3kaw==",
        "VisibilityTimeout": 36000
  },
  {
    "Id": "SecondMessage",
        "ReceiptHandle": "AQEBkTUH...HifSnw==",
        "VisibilityTimeout": 36000
  }
]
```

출력:

```
{
  "Successful": [
    {
      "Id": "SecondMessage"
    },
    {
      "Id": "FirstMessage"
    }
```

```
  ]
}
```

- 자세한 API 내용은 명령 참조ChangeMessageVisibilityBatch의 섹션을 참조하세요. AWS CLI

### `change-message-visibility`

다음 코드 예시에서는 `change-message-visibility`을 사용하는 방법을 보여 줍니다.

AWS CLI

메시지의 제한 시간 가시성을 변경하는 방법

이 예시에서는 지정된 메시지의 제한 시간 가시성을 10시간(10시간 * 60분 * 60초)으로 변경합니다.

명령:

```
aws sqs change-message-visibility --queue-url https://sqs.us-
east-1.amazonaws.com/80398EXAMPLE/MyQueue --receipt-handle AQEBTpyI...t6HyQg== --
visibility-timeout 36000
```

출력:

```
None.
```

- 자세한 API 내용은 명령 참조ChangeMessageVisibility의 섹션을 참조하세요. AWS CLI

### `create-queue`

다음 코드 예시에서는 `create-queue`을 사용하는 방법을 보여 줍니다.

AWS CLI

대기열을 생성하려면

이 예시에서는 지정된 이름의 대기열을 만들고, 메시지 보존 기간을 3일(3일 * 24시간 * 60분 * 60초)로 설정하고, 대기열의 DLQ(Dead Letter Queue)를 최대 메시지 수신 개수 1,000개의 지정된 대기열로 설정합니다.

명령:

```
aws sqs create-queue --queue-name MyQueue --attributes file://create-queue.json
```

입력 파일(create-queue.json):

```
{
  "RedrivePolicy": "{\"deadLetterTargetArn\":\"arn:aws:sqs:us-
east-1:80398EXAMPLE:MyDeadLetterQueue\",\"maxReceiveCount\":\"1000\"}",
  "MessageRetentionPeriod": "259200"
}
```

출력:

```
{
  "QueueUrl": "https://queue.amazonaws.com/80398EXAMPLE/MyQueue"
}
```

- 자세한 API 내용은 명령 참조CreateQueue의 섹션을 참조하세요. AWS CLI

### `delete-message-batch`

다음 코드 예시에서는 `delete-message-batch`을 사용하는 방법을 보여 줍니다.

AWS CLI

여러 메시지를 배치로 삭제하는 방법

이 예시에서는 지정된 메시지를 삭제합니다.

명령:

```
aws sqs delete-message-batch --queue-url https://sqs.us-
east-1.amazonaws.com/80398EXAMPLE/MyQueue --entries file://delete-message-batch.json
```

입력 파일(delete-message-batch.json):

```
[
  {
        "Id": "FirstMessage",
        "ReceiptHandle": "AQEB1mgl...Z4GuLw=="
  },
```

```
  {
    "Id": "SecondMessage",
        "ReceiptHandle": "AQEBLsYM...VQubAA=="
  }
]
```

출력:

```
{
  "Successful": [
    {
      "Id": "FirstMessage"
    },
    {
      "Id": "SecondMessage"
    }
  ]
}
```

- 자세한 API 내용은 명령 참조DeleteMessageBatch의 섹션을 참조하세요. AWS CLI

### `delete-message`

다음 코드 예시에서는 `delete-message`을 사용하는 방법을 보여 줍니다.

AWS CLI

메시지를 삭제하는 방법

이 예시에서는 지정된 메시지를 삭제합니다.

명령:

```
aws sqs delete-message --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/
MyQueue --receipt-handle AQEBRXTo...q2doVA==
```

출력:

```
None.
```

- 자세한 API 내용은 명령 참조DeleteMessage의 섹션을 참조하세요. AWS CLI

### delete-queue

다음 코드 예시에서는 delete-queue을 사용하는 방법을 보여 줍니다.

AWS CLI

대기열을 삭제하려면

이 예시에서는 지정된 대기열을 삭제합니다.

명령:

```
aws sqs delete-queue --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyNewerQueue
```

출력:

```
None.
```

- 자세한 API 내용은 명령 참조DeleteQueue의 섹션을 참조하세요. AWS CLI

### get-queue-attributes

다음 코드 예시에서는 get-queue-attributes을 사용하는 방법을 보여 줍니다.

AWS CLI

대기열의 속성을 가져오는 방법

이 예시에서는 지정된 대기열의 모든 속성을 가져옵니다.

명령:

```
aws sqs get-queue-attributes --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyQueue --attribute-names All
```

출력:

```
{
  "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"
  }
}
```

이 예시에서는 지정된 대기열의 최대 메시지 크기 및 가시성 제한 시간 속성만 가져옵니다.

명령:

```
aws sqs get-queue-attributes --queue-url https://sqs.us-
east-1.amazonaws.com/80398EXAMPLE/MyNewQueue --attribute-
names MaximumMessageSize VisibilityTimeout
```

출력:

```
{
  "Attributes": {
    "VisibilityTimeout": "30",
    "MaximumMessageSize": "262144"
  }
}
```

- 자세한 API 내용은 명령 참조GetQueueAttributes의 섹션을 참조하세요. AWS CLI

### `get-queue-url`

다음 코드 예시에서는 `get-queue-url`을 사용하는 방법을 보여 줍니다.

AWS CLI

대기열을 가져오려면 URL

이 예제에서는 지정된 대기열의 를 가져옵니다URL.

명령:

```
aws sqs get-queue-url --queue-name MyQueue
```

출력:

```
{
  "QueueUrl": "https://queue.amazonaws.com/80398EXAMPLE/MyQueue"
}
```

- 자세한 API 내용은 명령 참조GetQueueUrl의 섹션을 참조하세요. AWS CLI

### `list-dead-letter-source-queues`

다음 코드 예시에서는 `list-dead-letter-source-queues`을 사용하는 방법을 보여 줍니다.

AWS CLI

데드레터 소스 대기열을 나열하려면

이 예제에서는 지정된 데드레터 소스 대기열과 연결된 대기열을 나열합니다.

명령:

```
aws sqs list-dead-letter-source-queues --queue-url https://sqs.us-
east-1.amazonaws.com/80398EXAMPLE/MyDeadLetterQueue
```

출력:

```
{
  "queueUrls": [
    "https://queue.amazonaws.com/80398EXAMPLE/MyQueue",
    "https://queue.amazonaws.com/80398EXAMPLE/MyOtherQueue"
  ]
}
```

- 자세한 API 내용은 명령 참조ListDeadLetterSourceQueues의 섹션을 참조하세요. AWS CLI

### list-message-move-tasks

다음 코드 예시에서는 list-message-move-tasks을 사용하는 방법을 보여 줍니다.

AWS CLI

메시지 이동 작업을 나열하려면

다음 list-message-move-tasks 예제에서는 지정된 대기열에서 가장 최근의 메시지 이동 작업 2개를 나열합니다.

```
aws sqs list-message-move-tasks \
    --source-arn arn:aws:sqs:us-west-2:80398EXAMPLE:MyQueue \
    --max-results 2
```

출력:

```
{
    "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
         }
    ]
}
```

자세한 내용은 개발자 안내서의 Amazon SQS API 권한: 작업 및 리소스 참조를 참조하세요.

- 자세한 API 내용은 명령 참조ListMessageMoveTasks의 섹션을 참조하세요. AWS CLI

### `list-queue-tags`

다음 코드 예시에서는 `list-queue-tags`을 사용하는 방법을 보여 줍니다.

AWS CLI

대기열에 대한 모든 비용 할당 태그를 나열하려면

다음 `list-queue-tags` 예제에서는 지정된 대기열과 연결된 모든 비용 할당 태그를 표시합니다.

```
aws sqs list-queue-tags \
    --queue-url https://sqs.us-west-2.amazonaws.com/123456789012/MyQueue
```

출력:

```
{
    "Tags": {
        "Team": "Alpha"
    }
}
```

자세한 내용은 Amazon Simple Queue Service 개발자 안내서의 비용 할당 태그 나열을 참조하세요.

- 자세한 API 내용은 명령 참조ListQueueTags의 섹션을 참조하세요. AWS CLI

### `list-queues`

다음 코드 예시에서는 `list-queues`을 사용하는 방법을 보여 줍니다.

AWS CLI

대기열을 나열하는 방법

이 예시에서는 모든 대기열을 나열합니다.

명령:

```
aws sqs list-queues
```

출력:

```
{
```

```
  "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"
  ]
}
```

이 예시에서는 'My'로 시작하는 대기열만 나열합니다.

명령:

```
aws sqs list-queues --queue-name-prefix My
```

출력:

```
{
  "QueueUrls": [
    "https://queue.amazonaws.com/80398EXAMPLE/MyDeadLetterQueue",
    "https://queue.amazonaws.com/80398EXAMPLE/MyQueue",
    "https://queue.amazonaws.com/80398EXAMPLE/MyOtherQueue"
  ]
}
```

- 자세한 API 내용은 명령 참조ListQueues의 섹션을 참조하세요. AWS CLI

### purge-queue

다음 코드 예시에서는 purge-queue을 사용하는 방법을 보여 줍니다.

AWS CLI

대기열을 제거하려면

이 예제에서는 지정된 대기열의 모든 메시지를 삭제합니다.

명령:

```
aws sqs purge-queue --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/
MyNewQueue
```

출력:

```
None.
```

- 자세한 API 내용은 명령 참조PurgeQueue의 섹션을 참조하세요. AWS CLI

### `receive-message`

다음 코드 예시에서는 `receive-message`을 사용하는 방법을 보여 줍니다.

AWS CLI

메시지를 수신하는 방법

이 예시에서는 사용 가능한 메시지를 최대 10개까지 수신하고 사용 가능한 속성을 모두 반환합니다.

명령:

```
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
```

출력:

```
{
  "Messages": [
    {
      "Body": "My first message.",
      "ReceiptHandle": "AQEBzbVv...fqNzFw==",
      "MD5OfBody": "1000f835...a35411fa",
      "MD5OfMessageAttributes": "9424c491...26bc3ae7",
      "MessageId": "d6790f8d-d575-4f01-bc51-40122EXAMPLE",
      "Attributes": {
        "ApproximateFirstReceiveTimestamp": "1442428276921",
        "SenderId": "AIDAIAZKMSNQ7TEXAMPLE",
        "ApproximateReceiveCount": "5",
        "SentTimestamp": "1442428276921"
      },
      "MessageAttributes": {
        "PostalCode": {
```

```
          "DataType": "String",
          "StringValue": "ABC123"
        },
        "City": {
          "DataType": "String",
          "StringValue": "Any City"
        }
      }
    }
  ]
}
```

이 예제는 메시지 SentTimestamp 속성뿐만 아니라 SenderId 및 속성만 반환하는 다음 사용 가능한 PostalCode 메시지를 수신합니다.

명령:

```
aws sqs receive-message --queue-url https://sqs.us-
east-1.amazonaws.com/80398EXAMPLE/MyQueue --attribute-names SenderId SentTimestamp
 --message-attribute-names PostalCode
```

출력:

```
{
  "Messages": [
    {
      "Body": "My first message.",
      "ReceiptHandle": "AQEB6nR4...HzlvZQ==",
      "MD5OfBody": "1000f835...a35411fa",
      "MD5OfMessageAttributes": "b8e89563...e088e74f",
      "MessageId": "d6790f8d-d575-4f01-bc51-40122EXAMPLE",
      "Attributes": {
        "SenderId": "AIDAIAZKMSNQ7TEXAMPLE",
        "SentTimestamp": "1442428276921"
      },
      "MessageAttributes": {
        "PostalCode": {
          "DataType": "String",
          "StringValue": "ABC123"
        }
      }
    }
  ]
```

```
}
```

- 자세한 API 내용은 명령 참조ReceiveMessage의 섹션을 참조하세요. AWS CLI

### `remove-permission`

다음 코드 예시에서는 `remove-permission`을 사용하는 방법을 보여 줍니다.

AWS CLI

권한을 제거하려면

이 예제에서는 지정된 대기열에서 지정된 레이블이 있는 권한을 제거합니다.

명령:

```
aws sqs remove-permission --queue-url https://sqs.us-
east-1.amazonaws.com/80398EXAMPLE/MyQueue --label SendMessagesFromMyQueue
```

출력:

```
None.
```

- 자세한 API 내용은 명령 참조RemovePermission의 섹션을 참조하세요. AWS CLI

### `send-message-batch`

다음 코드 예시에서는 `send-message-batch`을 사용하는 방법을 보여 줍니다.

AWS CLI

여러 메시지를 배치로 전송하는 방법

이 예시에서는 지정된 메시지 본문, 지연 기간 및 메시지 속성이 설정된 메시지 2개를 지정된 대기열로 보냅니다.

명령:

```
aws sqs send-message-batch --queue-url https://sqs.us-
east-1.amazonaws.com/80398EXAMPLE/MyQueue --entries file://send-message-batch.json
```

입력 파일(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"
        }
          }
    }
]
```

출력:

```
{
  "Successful": [
    {
      "MD5OfMessageBody": "203c4a38...7943237e",
      "MD5OfMessageAttributes": "10809b55...baf283ef",
      "Id": "FuelReport-0001-2015-09-16T140731Z",
      "MessageId": "d175070c-d6b8-4101-861d-adeb3EXAMPLE"
    },
    {
      "MD5OfMessageBody": "2cf0159a...c1980595",
      "MD5OfMessageAttributes": "55623928...ae354a25",
      "Id": "FuelReport-0002-2015-09-16T140930Z",
      "MessageId": "f9b7d55d-0570-413e-b9c5-a9264EXAMPLE"
    }
  ]
}
```

- 자세한 API 내용은 명령 참조SendMessageBatch의 섹션을 참조하세요. AWS CLI

### `send-message`

다음 코드 예시에서는 `send-message`을 사용하는 방법을 보여 줍니다.

AWS CLI

메시지를 전송하려면

이 예시에서는 지정된 메시지 본문, 지연 기간 및 메시지 속성이 설정된 메시지를 지정된 대기열로 보냅니다.

명령:

```
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
```

입력 파일(send-message.json):

```
{
  "City": {
    "DataType": "String",
    "StringValue": "Any City"
  },
  "Greeting": {
    "DataType": "Binary",
    "BinaryValue": "Hello, World!"
  },
  "Population": {
    "DataType": "Number",
    "StringValue": "1250800"
  }
}
```

출력:

```
{
  "MD5OfMessageBody": "51b0a325...39163aa0",
  "MD5OfMessageAttributes": "00484c68...59e48f06",
  "MessageId": "da68f62c-0c07-4bee-bf5f-7e856EXAMPLE"
}
```

- 자세한 API 내용은 명령 참조SendMessage의 섹션을 참조하세요. AWS CLI

### `set-queue-attributes`

다음 코드 예시에서는 `set-queue-attributes`을 사용하는 방법을 보여 줍니다.

AWS CLI

대기열 속성을 설정하는 방법

이 예시에서는 지정된 대기열을 전송 지연 10초, 최대 메시지 크기 128KB(128KB * 1,024바이트), 메시지 보존 기간 3일(3일 * 24시간 * 60분 * 60초), 메시지 수신 대기 시간 20초, 기본 가시성 제한 시간 60초로 설정합니다. 또한 이 예시에서는 지정된 DLQ(Dead Letter Queue)를 최대 수신 개수 1,000개의 메시지와 연결합니다.

명령:

```
aws sqs set-queue-attributes --queue-url https://sqs.us-east-1.amazonaws.com/80398EXAMPLE/MyNewQueue --attributes file://set-queue-attributes.json
```

입력 파일(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"
}
```

출력:

```
None.
```

- 자세한 API 내용은 명령 참조SetQueueAttributes의 섹션을 참조하세요. AWS CLI

### `start-message-move-task`

다음 코드 예시에서는 `start-message-move-task`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: *메시지 이동 작업을 시작하려면*

다음 start-message-move-task 예제에서는 메시지 이동 작업을 시작하여 지정된 데드 레터 대기열에서 소스 대기열로 메시지를 재구동합니다.

```
aws sqs start-message-move-task \
    --source-arn arn:aws:sqs:us-west-2:80398EXAMPLE:MyQueue
```

출력:

```
{
    "TaskHandle": "AQEB6nR4...HzlvZQ=="
}
```

자세한 내용은 가이드 이름 의 주제 제목을 참조하세요.

예제 2: *메시지 이동 작업을 최대 속도로 시작하려면*

다음 start-message-move-task 예제에서는 메시지 이동 작업을 시작하여 지정된 데드 레터 대기열에서 지정된 대상 대기열로 메시지를 초당 최대 50개의 메시지 속도로 재구동합니다.

```
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
```

출력:

```
{
    "TaskHandle": "AQEB6nR4...HzlvZQ=="
}
```

자세한 내용은 개발자 안내서의 Amazon SQS API 권한: 작업 및 리소스 참조를 참조하세요.

- 자세한 API 내용은 명령 참조StartMessageMoveTask의 섹션을 참조하세요. AWS CLI

### tag-queue

다음 코드 예시에서는 tag-queue을 사용하는 방법을 보여 줍니다.

AWS CLI

대기열에 비용 할당 태그를 추가하려면

다음 tag-queue 예제에서는 지정된 Amazon SQS 대기열에 비용 할당 태그를 추가합니다.

```
aws sqs tag-queue \
    --queue-url https://sqs.us-west-2.amazonaws.com/123456789012/MyQueue \
    --tags Priority=Highest
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Simple Queue Service 개발자 안내서의 비용 할당 태그 추가를 참조하세요.

- 자세한 API 내용은 명령 참조TagQueue의 섹션을 참조하세요. AWS CLI

**untag-queue**

다음 코드 예시에서는 untag-queue을 사용하는 방법을 보여 줍니다.

AWS CLI

대기열에서 비용 할당 태그를 제거하려면

다음 untag-queue 예제에서는 지정된 Amazon SQS 대기열에서 비용 할당 태그를 제거합니다.

```
aws sqs untag-queue \
    --queue-url https://sqs.us-west-2.amazonaws.com/123456789012/MyQueue \
    --tag-keys "Priority"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Simple Queue Service 개발자 안내서의 비용 할당 태그 추가를 참조하세요.

- 자세한 API 내용은 명령 참조UntagQueue의 섹션을 참조하세요. AWS CLI

## 를 사용한 Storage Gateway 예제 AWS CLI

다음 코드 예제에서는 Storage Gateway 와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `describe-gateway-information`

다음 코드 예시에서는 `describe-gateway-information`을 사용하는 방법을 보여 줍니다.

AWS CLI

게이트웨이를 설명하려면

다음 `describe-gateway-information` 명령은 지정된 게이트웨이에 대한 메타데이터를 반환합니다. 설명할 게이트웨이를 지정하려면 명령에서 게이트웨이의 Amazon 리소스 이름(ARN)을 사용합니다.

이 예제에서는 계정 `sgw-12A3456B` 에 ID가 있는 게이트웨이를 지정합니다`123456789012`.

```
aws storagegateway describe-gateway-information --gateway-
arn "arn:aws:storagegateway:us-west-2:123456789012:gateway/sgw-12A3456B"
```

이 명령은 이름, 네트워크 인터페이스, 구성된 시간대 및 상태(게이트웨이가 실행 중인지 여부)와 같은 게이트웨이에 대한 메타데이터가 포함된 JSON 블록을 출력합니다.

- 자세한 API 내용은 명령 참조DescribeGatewayInformation의 섹션을 참조하세요. AWS CLI

### `list-file-shares`

다음 코드 예시에서는 `list-file-shares`을 사용하는 방법을 보여 줍니다.

AWS CLI

파일 공유를 나열하려면

다음 `command-name` 예제에서는 AWS 계정에서 사용 가능한 위젯을 나열합니다.

```
aws storagegateway list-file-shares \
    --gateway-arn arn:aws:storagegateway:us-east-1:209870788375:gateway/sgw-FB02E292
```

출력:

```
{
    "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
}
```

자세한 내용은 Storage Gateway 서비스 참조ListFileShares의 섹션을 참조하세요. AWS Storage Gateway API

- 자세한 API 내용은 명령 참조ListFileShares의 섹션을 참조하세요. AWS CLI

### `list-gateways`

다음 코드 예시에서는 `list-gateways`을 사용하는 방법을 보여 줍니다.

AWS CLI

계정의 게이트웨이를 나열하려면

다음 `list-gateways` 명령은 계정에 정의된 모든 게이트웨이를 나열합니다.

```
aws storagegateway list-gateways
```

이 명령은 게이트웨이 Amazon 리소스 이름() 목록이 포함된 JSON 블록을 출력합니다ARNs.

- 자세한 API 내용은 명령 참조ListGateways의 섹션을 참조하세요. AWS CLI

### `list-volumes`

다음 코드 예시에서는 `list-volumes`을 사용하는 방법을 보여 줍니다.

AWS CLI

게이트웨이에 대해 구성된 볼륨을 나열하려면

다음 `list-volumes` 명령은 지정된 게이트웨이에 대해 구성된 볼륨 목록을 반환합니다. 설명할 게이트웨이를 지정하려면 명령에서 게이트웨이의 Amazon 리소스 이름(ARN)을 사용합니다.

이 예제에서는 계정 `sgw-12A3456B` 에 ID가 있는 게이트웨이를 지정합니다`123456789012`.

```
aws storagegateway list-volumes --gateway-arn "arn:aws:storagegateway:us-
west-2:123456789012:gateway/sgw-12A3456B"
```

이 명령은 각 볼륨에 ARN 대해 유형 및 를 포함하는 볼륨 목록을 출력하는 JSON 블록을 출력합니다.

- 자세한 API 내용은 명령 참조ListVolumes의 섹션을 참조하세요. AWS CLI

### `refresh-cache`

다음 코드 예시에서는 `refresh-cache`을 사용하는 방법을 보여 줍니다.

AWS CLI

파일 공유 캐시를 새로 고치려면

다음 `refresh-cache` 예제에서는 지정된 파일 공유의 캐시를 새로 고칩니다.

```
aws storagegateway refresh-cache \
    --file-share-arn arn:aws:storagegateway:us-east-1:111122223333:share/
share-2FA12345
```

출력:

```
{
    "FileShareARN": "arn:aws:storagegateway:us-east-1:111122223333:share/
share-2FA12345",
```

```
    "NotificationId": "4954d4b1-abcd-ef01-1234-97950a7d3483"
}
```

자세한 내용은 Storage Gateway 서비스 참조ListFileShares의 섹션을 참조하세요. AWS Storage Gateway API

- 자세한 API 내용은 명령 참조RefreshCache의 섹션을 참조하세요. AWS CLI

## AWS STS 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS STS.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `assume-role-with-saml`

다음 코드 예시에서는 `assume-role-with-saml`을 사용하는 방법을 보여 줍니다.

AWS CLI

로 인증된 역할에 대한 단기 자격 증명을 가져오려면 SAML

다음 `assume-role-with-saml` 명령은 IAM 역할 에 대한 단기 자격 증명 세트를 검색합니다`TestSaml`. 이 예제의 요청은 인증 시 자격 증명 공급자가 제공한 SAML어설션을 사용하여 인증됩니다.

```
aws sts assume-role-with-saml \
    --role-arn arn:aws:iam::123456789012:role/TestSaml \
```

```
    --principal-arn arn:aws:iam::123456789012:saml-provider/SAML-test \
    --saml-
assertion "VERYLONGENCODEDASSERTIONEXAMPLExzYW1sOkF1ZGllbmNlPmJsYW5rPC9zYW1sOkF1ZGllbmNlPjwv
+PHNhbWw6TmFtZUlEIEZvcm1hdD0idXJuOm9hc2lzOm5hbWVzOnRjOlNBTUw6Mi4wOm5hbWVpZC1mb3JtYXQ6dHJhbnN
+PHNhbWw6U3ViamVjdENvbmZpcm1hdGlvbiBNZXRob2Q9InVybjpvYXNpczpuYW1lczp0YzpTQU1MOjIuMDpjbTpiZWF
```

출력:

```
{
    "Issuer": "https://integ.example.com/idp/shibboleth</Issuer",
    "AssumedRoleUser": {
        "Arn": "arn:aws:sts::123456789012:assumed-role/TestSaml",
        "AssumedRoleId": "ARO456EXAMPLE789:TestSaml"
    },
    "Credentials": {
        "AccessKeyId": "ASIAV3ZUEFP6EXAMPLE",
        "SecretAccessKey": "8P+SQvWIuLnKhh8d++jpw0nNmQRBZvNEXAMPLEKEY",
        "SessionToken": "IQoJb3JpZ2luX2VjEOz////////////////////
wEXAMPLEtMSJHMEUCIDoKK3JH9uGQE1z0sINr5M4jk
+Na8KHDcCYRVjJCZEvOAiEA3OvJGtw1EcViOleS2vhs8VdCKFJQWPQrmGdeehM4IC1NtBmUpp2wUE8phUZampKsburED
+xo0rKwT38xVqr7ZD0u0iPPkUL64lIZbqBAz
+scqKmlzm8FDrypNC9Yjc8fPOLn9FX9KSYvKTr4rvx3iSIlTJabIQwj2ICCR/oLxBA==",
        "Expiration": "2019-11-01T20:26:47Z"
    },
    "Audience": "https://signin.aws.amazon.com/saml",
    "SubjectType": "transient",
    "PackedPolicySize": "6",
    "NameQualifier": "SbdGOnUkh1i4+EXAMPLExL/jEvs=",
    "Subject": "SamlExample"
}
```

자세한 내용은 AWS IAM 사용 설명서의 임시 보안 자격 증명 요청을 참조하세요.

- 자세한 API 내용은 명령 참조AssumeRoleWithSaml의 섹션을 참조하세요. AWS CLI

### `assume-role-with-web-identity`

다음 코드 예시에서는 `assume-role-with-web-identity`을 사용하는 방법을 보여 줍니다.

AWS CLI

Web Identity로 인증된 역할에 대한 단기 자격 증명을 가져오려면(OAuth 2.'0)

다음 assume-role-with-web-identity 명령은 IAM 역할 에 대한 단기 자격 증명 세트를 검색합니다app1. 요청은 지정된 웹 ID 제공업체가 제공하는 웹 ID 토큰을 사용하여 인증됩니다. 사용자가 수행할 수 있는 작업을 추가로 제한하기 위해 두 가지 추가 정책이 세션에 적용됩니다. 반환된 자격 증명은 생성되고 1시간 후에 만료됩니다.

```
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
%7CIQEBLjAsAhRFiXuWpUXuRvQ9PZL3GMFcYevydwIUFAHZwXZXXXXXXXXJnrulxKDHwy87oGKPznh0D6bEQZTSCzyoC
CrKqjG7nPBjNIL016GGvuS5gSvPRUxWES3VYfm1wl7WTI7jn-Pcb6M-
buCgHhFOzTQxod27L9CqnOLio7N3gZAGpsp6n1-
AJBOCJckcyXe2c6uD0srOJeZlKUm2eTDVMf8IehDVI0r1QOnTV6KzzAI3OY87Vd_cVMQ"
```

출력:

```
{
    "SubjectFromWebIdentityToken": "amzn1.account.AF6RHO7KZU5XRVQJGXK6HB56KR2A"
    "Audience": "client.5498841531868486423.1548@apps.example.com",
    "AssumedRoleUser": {
        "Arn": "arn:aws:sts::123456789012:assumed-role/FederatedWebIdentityRole/
app1",
        "AssumedRoleId": "AROACLKWSDQRAOEXAMPLE:app1"
    }
    "Credentials": {
        "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYzEXAMPLEKEY",
        "SessionToken": "AQoEXAMPLEH4aoAH0gNCAPyJxz4BlCFFxWNE1OPTgk5TthT
+FvwqnKwRcOIfrRh3c/LTo6UDdyJwOOvEVPvLXCrrrUtdnniCEXAMPLE/
IvU1dYUg2RVAJBanLiHb4IgRmpRV3zrkuWJOgQs8IZZaIv2BXIa2R4OlgkBN9bkUDNCJiBeb/
AXlzBBko7b15fjrBs2+cTQtpZ3CYWFXG8C5zqx37wnOE49mRl/+OtkIKGO7fAE",
        "Expiration": "2020-05-19T18:06:10+00:00"
    },
    "Provider": "www.amazon.com"
}
```

자세한 내용은 AWS IAM 사용 설명서의 임시 보안 자격 증명 요청을 참조하세요.

- 자세한 API 내용은 명령 참조AssumeRoleWithWebIdentity의 섹션을 참조하세요. AWS CLI

### assume-role

다음 코드 예시에서는 assume-role을 사용하는 방법을 보여 줍니다.

AWS CLI

역할 위임

다음 assume-role 명령은 IAM 역할 에 대한 단기 자격 증명 세트를 검색합니다s3-access-example.

```
aws sts assume-role \
    --role-arn arn:aws:iam::123456789012:role/xaccounts3access \
    --role-session-name s3-access-example
```

출력:

```
{
    "AssumedRoleUser": {
        "AssumedRoleId": "AROA3XFRBF535PLBIFPI4:s3-access-example",
        "Arn": "arn:aws:sts::123456789012:assumed-role/xaccounts3access/s3-access-
example"
    },
    "Credentials": {
        "SecretAccessKey": "9drTJvcXLB89EXAMPLELB8923FB892xMFI",
        "SessionToken": "AQoXdzELDDY//////////
wEaoAK1wvxJY12r2IrDFT2IvAzTCn3zHoZ7YNtpiQLF0MqZye/qwjzP2iEXAMPLEbw/
m3hsj8VBTkPORGvr9jM5sgP+w9IZWZnU+LWhmg
+a5fDi2oTGUYcdg9uexQ4mtCHIHfi4citgqZTgco40Yqr4lIlo4V2b2Dyauk0eYFNebHtYlFVgAUj
+7Indz3LU0aTWk1WKIjHmmMCIoTkyYp/k7kUG7moeEYKSitwQIi6Gjn+nyzM
+PtoA3685ixzv0R7i5rjQi0YE0lf1oeie3bDiNHncmzosRM6SFiPzSvp6h/32xQuZsjcypmwsPSDtTPYcs0+YN/8BRi2
IcrxSpnWEXAMPLEXSDFTAQAM6Dl9zR0tXoybnlrZIwMLlMi1Kcgo5OytwU=",
        "Expiration": "2016-03-15T00:05:07Z",
        "AccessKeyId": "ASIAJEXAMPLEXEG2JICEA"
    }
}
```

명령의 출력에는 AWS인증에 사용할 수 있는 액세스 키, 비밀 키 및 세션 토큰이 포함됩니다.

사용을 위해 AWS CLI 역할과 연결된 명명된 프로파일을 설정할 수 있습니다. 프로필을 사용하면 AWS CLI가 수임 역할을 호출하고 자격 증명을 관리합니다. 자세한 내용은 AWS CLI 사용 설명서의 에서 IAM 역할 사용을 AWS CLI 참조하세요.

- 자세한 API 내용은 명령 참조AssumeRole의 섹션을 참조하세요. AWS CLI

### decode-authorization-message

다음 코드 예시에서는 decode-authorization-message을 사용하는 방법을 보여 줍니다.

AWS CLI

요청에 대한 응답으로 반환된 인코딩된 인증 메시지 디코딩

다음 decode-authorization-message 예제는 Amazon Web Services 요청에 대한 응답으로 반환되는 인코딩 메시지로부터 받은 요청의 권한 부여 상태에 대한 추가 정보를 디코딩합니다.

```
aws sts decode-authorization-message \
    --encoded-message EXAMPLEWodyRNrtlQARDip-
eTA6i6DrlUhHhPQrLWB_lAbl5pAKxl9mPDLexYcGBreyIKQC1BGBIpBKr3dFDkwqeO7e2NMk5j_hmzAiChJN-8oy3Ewi
Ojau7BMjOTWwOtHPhV_Zaz87yENdipr745EjQwRd5LaoL3vN8_5ZfA9UiBMKDgVh1gjqZJFUiQoubv78V1RbHNYnK44E
pOu3FZjwYStfvTb3GHs3-6rLribGO9jZOktkfE6vqxlFzLyeDr4P2ihC1wty9tArCvvGzIAUNmARQJ2VVWPxioqgoqCz
JWP5pwe_mAyqh0NLw-r1S56YC_90onj9A80sNrHlI-
tIiNd7tgNTYzDuPQYD2FMDBnp82V9eVmYGtPp5NIeSpuf3fOHanFuBZgENxZQZ2dlH3xJGMTtYayzZrRXjiq_SfX9zeB
FaoPIb8LmmKVBLpIB0iFhU9sEHPqKHVPi6jdxXqKaZaFGvYVmVOiuQdNQKuyk0p067POFrZECLjjOtNPBOZCcuEKEXAM
```

출력:

```
{
    "DecodedMessage": "{\"allowed\":false,\"explicitDeny\":true,\"matchedStatements
\":{\"items\":[{\"statementId\":\"VisualEditor0\",\"effect\":\"DENY\",\"principals
\":{\"items\":[{\"value\":\"AROA123456789EXAMPLE\"}]},\"principalGroups
\":{\"items\":[]},\"actions\":{\"items\":[{\"value\":\"ec2:RunInstances
\"}]},\"resources\":{\"items\":[{\"value\":\"*\"}]},\"conditions\":{\"items
\":[]}}]},\"failures\":{\"items\":[]},\"context\":{\"principal\":{\"id\":
\"AROA123456789EXAMPLE:Ana\",\"arn\":\"arn:aws:sts::111122223333:assumed-role/
Developer/Ana\"},\"action\":\"RunInstances\",\"resource\":\"arn:aws:ec2:us-
east-1:111122223333:instance/*\",\"conditions\":{\"items\":[{\"key\":
\"ec2:MetadataHttpPutResponseHopLimit\",\"values\":{\"items\":[{\"value\":
\"2\"}]}},{\"key\":\"ec2:InstanceMarketType\",\"values\":{\"items\":[{\"value
\":\"on-demand\"}]}},{\"key\":\"aws:Resource\",\"values\":{\"items\":[{\"value
\":\"instance/*\"}]}},{\"key\":\"aws:Account\",\"values\":{\"items\":[{\"value
\":\"111122223333\"}]}},{\"key\":\"ec2:AvailabilityZone\",\"values\":{\"items\":
```

```
[{\"value\":\"us-east-1f\"}]}},{\"key\":\"ec2:ebsOptimized\",\"values\":{\"items
\":[{\"value\":\"false\"}]}},{\"key\":\"ec2:IsLaunchTemplateResource\",\"values
\":{\"items\":[{\"value\":\"false\"}]}},{\"key\":\"ec2:InstanceType\",\"values\":
{\"items\":[{\"value\":\"t2.micro\"}]}},{\"key\":\"ec2:RootDeviceType\",\"values
\":{\"items\":[{\"value\":\"ebs\"}]}},{\"key\":\"aws:Region\",\"values\":{\"items
\":[{\"value\":\"us-east-1\"}]}},{\"key\":\"ec2:MetadataHttpEndpoint\",\"values
\":{\"items\":[{\"value\":\"enabled\"}]}},{\"key\":\"aws:Service\",\"values\":
{\"items\":[{\"value\":\"ec2\"}]}},{\"key\":\"ec2:InstanceID\",\"values\":{\"items
\":[{\"value\":\"*\"}]}},{\"key\":\"ec2:MetadataHttpTokens\",\"values\":{\"items
\":[{\"value\":\"required\"}]}},{\"key\":\"aws:Type\",\"values\":{\"items\":
[{\"value\":\"instance\"}]}},{\"key\":\"ec2:Tenancy\",\"values\":{\"items\":
[{\"value\":\"default\"}]}},{\"key\":\"ec2:Region\",\"values\":{\"items\":[{\"value
\":\"us-east-1\"}]}},{\"key\":\"aws:ARN\",\"values\":{\"items\":[{\"value\":
\"arn:aws:ec2:us-east-1:111122223333:instance/*\"}]}}]}}}"
}
```

자세한 내용은 AWS IAM 사용 설명서의 정책 평가 로직을 참조하세요.

- 자세한 API 내용은 명령 참조DecodeAuthorizationMessage의 섹션을 참조하세요. AWS CLI

### `get-caller-identity`

다음 코드 예시에서는 `get-caller-identity`을 사용하는 방법을 보여 줍니다.

AWS CLI

현재 IAM 자격 증명에 대한 세부 정보를 가져오려면

다음 `get-caller-identity` 명령은 요청을 인증하는 데 사용되는 IAM 자격 증명에 대한 정보를 표시합니다. 발신자는 IAM 사용자입니다.

```
aws sts get-caller-identity
```

출력:

```
{
    "UserId": "AIDASAMPLEUSERID",
    "Account": "123456789012",
    "Arn": "arn:aws:iam::123456789012:user/DevAdmin"
}
```

- 자세한 API 내용은 명령 참조GetCallerIdentity의 섹션을 참조하세요. AWS CLI

### get-federation-token

다음 코드 예시에서는 get-federation-token을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 사용자 액세스 키 자격 증명을 사용하여 임시 보안 자격 증명 세트를 반환하려면

다음 get-federation-token 예제는 사용자에 대한 임시 보안 자격 증명 세트(액세스 키 ID, 비밀 액세스 키 및 보안 토큰으로 구성)를 반환합니다. IAM 사용자의 장기 보안 자격 증명을 사용하여 GetFederationToken 작업을 호출해야 합니다.

```
aws sts get-federation-token \
    --name Bob \
    --policy file://myfile.json \
    --policy-arns arn=arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess \
    --duration-seconds 900
```

myfile.json의 콘텐츠:

```
{
    "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": "*"
        }
    ]
}
```

출력:

```
{
    "Credentials": {
        "AccessKeyId": "ASIAIOSFODNN7EXAMPLE",
        "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
        "SessionToken": "EXAMPLEpZ2luX2VjEGoaCXVzLXdlc3QtMiJIMEYCIQC/
W9pL5ArQyDD5JwFL3/h5+WGopQ24GEXweNctwhi9sgIhAMkg
+MZE35iWM8s4r5Lr25f9rSTVPFH98G42QQunWMTfKq0DCOP/////////
wEQAxoMNDUyOTI1MTcwNTA3Igxuy3AOpuuoLsk3MJwqgQPg8QOd9HuoClUxq26wnc/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
}
```

자세한 내용은 AWS IAM 사용 설명서의 임시 보안 자격 증명 요청을 참조하세요.

- 자세한 API 내용은 명령 참조GetFederationToken의 섹션을 참조하세요. AWS CLI

### `get-session-token`

다음 코드 예시에서는 `get-session-token`을 사용하는 방법을 보여 줍니다.

AWS CLI

IAM 자격 증명에 대한 단기 자격 증명 세트를 가져오려면

다음 `get-session-token` 명령은 호출하는 IAM 자격 증명에 대한 단기 자격 증명 세트를 검색합니다. 정책에 따라 다중 인증(MFA)이 필요한 요청에 결과 자격 증명을 사용할 수 있습니다. 보안 인증 정보는 생성되고 15분 후에 만료됩니다.

```
aws sts get-session-token \
    --duration-seconds 900 \
    --serial-number "YourMFADeviceSerialNumber" \
    --token-code 123456
```

출력:

```
{
    "Credentials": {
        "AccessKeyId": "ASIAIOSFODNN7EXAMPLE",
        "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYzEXAMPLEKEY",
        "SessionToken": "AQoEXAMPLEH4aoAH0gNCAPyJxz4BlCFFxWNE1OPTgk5TthT
+FvwqnKwRcOIfrRh3c/LTo6UDdyJwOOvEVPvLXCrrrUtdnniCEXAMPLE/
IvU1dYUg2RVAJBanLiHb4IgRmpRV3zrkuWJOgQs8IZZaIv2BXIa2R4OlgkBN9bkUDNCJiBeb/
AXlzBBko7b15fjrBs2+cTQtpZ3CYWFXG8C5zqx37wnOE49mRl/+OtkIKGO7fAE",
        "Expiration": "2020-05-19T18:06:10+00:00"
    }
}
```

자세한 내용은 AWS IAM 사용 설명서의 임시 보안 자격 증명 요청을 참조하세요.

- 자세한 API 내용은 명령 참조GetSessionToken의 섹션을 참조하세요. AWS CLI

## AWS Support 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS Support.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### add-attachments-to-set

다음 코드 예시에서는 add-attachments-to-set을 사용하는 방법을 보여 줍니다.

AWS CLI

세트에 첨부 파일을 추가하려면

다음 add-attachments-to-set 예제에서는 AWS 계정의 지원 사례에 대해 지정할 수 있는 세트에 이미지를 추가합니다.

```
aws support add-attachments-to-set \
    --attachment-set-id "as-2f5a6faa2a4a1e600-mu-nk5xQlBr70-
G1cUos5LZkd38KOAHZa9BMDVzNEXAMPLE" \
    --attachments fileName=troubleshoot-screenshot.png,data=base64-encoded-string
```

출력:

```
{
    "attachmentSetId": "as-2f5a6faa2a4a1e600-mu-nk5xQlBr70-
G1cUos5LZkd38KOAHZa9BMDVzNEXAMPLE",
    "expiryTime": "2020-05-14T17:04:40.790+0000"
}
```

자세한 내용은 AWS Support 사용 설명서의 사례 관리를 참조하세요.

- 자세한 API 내용은 명령 참조AddAttachmentsToSet의 섹션을 참조하세요. AWS CLI

### add-communication-to-case

다음 코드 예시에서는 add-communication-to-case을 사용하는 방법을 보여 줍니다.

AWS CLI

사례에 통신을 추가하려면

다음 add-communication-to-case 예제에서는 AWS 계정의 지원 사례에 통신을 추가합니다.

```
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-
G1cUos5LZkd38KOAHZa9BMDVzNEXAMPLE"
```

출력:

```
{
    "result": true
}
```

자세한 내용은 AWS Support 사용 설명서의 사례 관리를 참조하세요.

- 자세한 API 내용은 명령 참조AddCommunicationToCase의 섹션을 참조하세요. AWS CLI

### `create-case`

다음 코드 예시에서는 `create-case`을 사용하는 방법을 보여 줍니다.

AWS CLI

사례를 생성하는 방법

다음 `create-case` 예제에서는 AWS 계정에 대한 지원 사례를 생성합니다.

```
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"
```

출력:

```
{
    "caseId": "case-12345678910-2013-c4c1d2bf33c5cf47"
}
```

자세한 내용은 AWS Support 사용 설명서의 사례 관리를 참조하세요.

- 자세한 API 내용은 명령 참조CreateCase의 섹션을 참조하세요. AWS CLI

### `describe-attachment`

다음 코드 예시에서는 `describe-attachment`을 사용하는 방법을 보여 줍니다.

AWS CLI

첨부 파일을 설명하는 방법

다음 `describe-attachment` 예시에서는 지정된 ID를 가진 첨부 파일에 대한 정보를 반환합니다.

```
aws support describe-attachment \
    --attachment-id "attachment-KBnjRNrePd9D6Jx0-Mm00xZuDEaL2JAj_0-
gJv9qqDooTipsz3V1Nb19rCfkZneeQeDPgp8X1iVJyHH7UuhZDdNeqGoduZsPrAhyMakqlc60-
iJjL5HqyYGiT1FG8EXAMPLE"
```

출력:

```
{
    "attachment": {
        "fileName": "troubleshoot-screenshot.png",
        "data": "base64-blob"
    }
}
```

자세한 내용은 AWS Support 사용 설명서의 사례 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeAttachment의 섹션을 참조하세요. AWS CLI

### `describe-cases`

다음 코드 예시에서는 `describe-cases`을 사용하는 방법을 보여 줍니다.

AWS CLI

사례를 설명하는 방법

다음 `describe-cases` 예제에서는 AWS 계정에서 지정된 지원 사례에 대한 정보를 반환합니다.

```
aws support describe-cases \
    --display-id "1234567890" \
```

```
    --after-time "2020-03-23T21:31:47.774Z" \
    --include-resolved-cases \
    --language "en" \
    --no-include-communications \
    --max-item 1
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS Support 사용 설명서의 사례 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeCases의 섹션을 참조하세요. AWS CLI

### describe-communications

다음 코드 예시에서는 describe-communications을 사용하는 방법을 보여 줍니다.

AWS CLI

사례에 대한 최신 커뮤니케이션을 설명하는 방법

다음 describe-communications 예제는 AWS 계정에서 지정된 지원 사례에 대한 최신 통신을 반환합니다.

```
aws support describe-communications \
```

```
    --case-id "case-12345678910-2013-c4c1d2bf33c5cf47" \
    --after-time "2020-03-23T21:31:47.774Z" \
    --max-item 1
```

출력:

```
{
    "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": "eyJuZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAxMDAwfQ==EXAMPLE=="
}
```

자세한 내용은 AWS Support 사용 설명서의 사례 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeCommunications의 섹션을 참조하세요. AWS CLI

### describe-services

다음 코드 예시에서는 describe-services을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 서비스 및 서비스 범주를 나열하려면

다음 describe-services 예시에서는 일반 정보를 요청하는 데 사용할 수 있는 서비스 범주를 나열합니다.

```
aws support describe-services \
    --service-code-list "general-info"
```

출력:

```
{
    "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"
                }
            ]
        }
    ]
}
```

자세한 내용은 AWS Support 사용 설명서의 사례 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeServices의 섹션을 참조하세요. AWS CLI

### describe-severity-levels

다음 코드 예시에서는 describe-severity-levels을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 심각도 수준을 나열하는 방법

다음 describe-severity-levels 예시에서는 지원 사례에 사용할 수 있는 심각도 수준을 나열합니다.

```
aws support describe-severity-levels
```

출력:

```
{
    "severityLevels": [
        {
            "code": "low",
            "name": "Low"
        },
        {
            "code": "normal",
            "name": "Normal"
        },
        {
            "code": "high",
            "name": "High"
        },
        {
            "code": "urgent",
            "name": "Urgent"
        },
        {
            "code": "critical",
            "name": "Critical"
        }
    ]
}
```

자세한 내용은 AWS Support 사용 설명서의 심각도 선택을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeSeverityLevels의 섹션을 참조하세요. AWS CLI

### describe-trusted-advisor-check-refresh-statuses

다음 코드 예시에서는 describe-trusted-advisor-check-refresh-statuses을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS Trusted Advisor 검사의 새로 고침 상태를 나열하려면

다음 describe-trusted-advisor-check-refresh-statuses 예제에서는 Amazon S3 버킷 권한 및 IAM 사용이라는 두 가지 Trusted Advisor 확인에 대한 새로 고침 상태를 나열합니다.

```
aws support describe-trusted-advisor-check-refresh-statuses \
    --check-id "Pfx0RwqBli" "zXCkfM1nI3"
```

출력:

```
{
    "statuses": [
        {
            "checkId": "Pfx0RwqBli",
            "status": "none",
            "millisUntilNextRefreshable": 0
        },
        {
            "checkId": "zXCkfM1nI3",
            "status": "none",
            "millisUntilNextRefreshable": 0
        }
    ]
}
```

자세한 내용은 AWS 지원 사용 설명서의 AWS Trusted Advisor를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeTrustedAdvisorCheckRefreshStatuses의 섹션을 참조하세요. AWS CLI

### describe-trusted-advisor-check-result

다음 코드 예시에서는 describe-trusted-advisor-check-result을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS Trusted Advisor 검사 결과를 나열하려면

다음 `describe-trusted-advisor-check-result` 예제에서는 IAM 사용 확인의 결과를 나열합니다.

```
aws support describe-trusted-advisor-check-result \
    --check-id "zXCkfM1nI3"
```

출력:

```
{
    "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
            }
        ]
    }
}
```

자세한 내용은 AWS 지원 사용 설명서의 AWS Trusted Advisor를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeTrustedAdvisorCheckResult의 섹션을 참조하세요. AWS CLI

### describe-trusted-advisor-check-summaries

다음 코드 예시에서는 describe-trusted-advisor-check-summaries을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS Trusted Advisor 검사 요약을 나열하려면

다음 describe-trusted-advisor-check-summaries 예제에서는 Amazon S3 버킷 권한 및 IAM 사용이라는 두 가지 Trusted Advisor 검사에 대한 결과를 나열합니다.

```
aws support describe-trusted-advisor-check-summaries \
    --check-ids "Pfx0RwqBli" "zXCkfM1nI3"
```

출력:

```
{
    "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
                }
            }
        }
    ]
}
```

자세한 내용은 AWS 지원 사용 설명서의 AWS Trusted Advisor를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeTrustedAdvisorCheckSummaries의 섹션을 참조하세요. AWS CLI

### describe-trusted-advisor-checks

다음 코드 예시에서는 describe-trusted-advisor-checks을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 AWS Trusted Advisor 검사를 나열하려면

다음 describe-trusted-advisor-checks 예제에서는 AWS 계정에서 사용 가능한 Trusted Advisor 검사를 나열합니다. 이 정보에는 수표 이름, ID, 설명, 범주 및 메타데이터가 포함됩니다. 가독성을 위해 출력이 단축됩니다.

```
aws support describe-trusted-advisor-checks \
    --language "en"
```

출력:

```
{
    "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": []
        }
    ]
}
```

자세한 내용은 AWS 지원 사용 설명서의 AWS Trusted Advisor를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeTrustedAdvisorChecks의 섹션을 참조하세요. AWS CLI

### `refresh-trusted-advisor-check`

다음 코드 예시에서는 `refresh-trusted-advisor-check`을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS Trusted Advisor 확인을 새로 고치려면

다음 `refresh-trusted-advisor-check` 예제에서는 AWS 계정의 Amazon S3 버킷 권한 Trusted Advisor 확인을 새로 고칩니다.

```
aws support refresh-trusted-advisor-check \
    --check-id "Pfx0RwqBli"
```

출력:

```
{
    "status": {
        "checkId": "Pfx0RwqBli",
        "status": "enqueued",
        "millisUntilNextRefreshable": 3599992
    }
```

```
}
```

자세한 내용은 AWS 지원 사용 설명서의 AWS Trusted Advisor를 참조하세요.

- 자세한 API 내용은 명령 참조RefreshTrustedAdvisorCheck의 섹션을 참조하세요. AWS CLI

**`resolve-case`**

다음 코드 예시에서는 `resolve-case`을 사용하는 방법을 보여 줍니다.

AWS CLI

지원 사례를 해결하는 방법

다음 `resolve-case` 예제에서는 AWS 계정의 지원 사례를 해결합니다.

```
aws support resolve-case \
    --case-id "case-12345678910-2013-c4c1d2bf33c5cf47"
```

출력:

```
{
    "finalCaseStatus": "resolved",
    "initialCaseStatus": "work-in-progress"
}
```

자세한 내용은 AWS Support 사용 설명서의 사례 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ResolveCase의 섹션을 참조하세요. AWS CLI

## 를 사용한 Amazon SWF 예제 AWS CLI

다음 코드 예제에서는 Amazon 와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다SWF.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `count-closed-workflow-executions`

다음 코드 예시에서는 `count-closed-workflow-executions`을 사용하는 방법을 보여 줍니다.

AWS CLI

종료된 워크플로 실행 수 계산

`swf count-closed-workflow-executions` 를 사용하여 지정된 도메인에 대해 종료된 워크플로 실행 수를 검색할 수 있습니다. 필터를 지정하여 특정 실행 클래스를 계산할 수 있습니다.

`--domain` 및 `--close-time-filter` 또는 `--start-time-filter` 인수가 필요합니다. 다른 모든 인수는 선택 사항입니다.

```
aws swf count-closed-workflow-executions \
    --domain DataFrobtzz \
    --close-time-filter "{ \"latestDate\" : 1377129600, \"oldestDate\" :
 1370044800 }"
```

출력:

```
{
    "count": 2,
    "truncated": false
}
```

'절단됨'이 `true`이면 'count'는 Amazon 에서 반환할 수 있는 최대 수를 나타냅니다SWF. 추가 결과는 잘립니다.

반환되는 결과 수를 줄이려면 다음을 수행할 수 있습니다.

`--close-time-filter` 또는 `--start-time-filter` 값을 수정하여 검색되는 시간 범위를 좁힙니다. 이 각각은 상호 배타적입니다. 요청에서 이 중 하나만 지정할 수 있습니다. `--close-status-filter`, `--execution-filter` `--tag-filter` 또는 `--type-filter` 인수를 사용하여 결과를 추가로 필터링합니다. 그러나 이러한 인수는 상호 배타적입니다.

Amazon Simple Workflow Service API 참조의 또한CountClosedWorkflowExecutions 참조

- 자세한 API 내용은 명령 참조CountClosedWorkflowExecutions의 섹션을 참조하세요. AWS CLI

### `count-open-workflow-executions`

다음 코드 예시에서는 `count-open-workflow-executions`을 사용하는 방법을 보여 줍니다.

AWS CLI

오픈 워크플로 실행 수 계산

`swf count-open-workflow-executions` 를 사용하여 지정된 도메인에 대해 열린 워크플로 실행 수를 검색할 수 있습니다. 필터를 지정하여 특정 실행 클래스를 계산할 수 있습니다.

`--domain` 및 `--start-time-filter` 인수가 필요합니다. 다른 모든 인수는 선택 사항입니다.

```
aws swf count-open-workflow-executions \
    --domain DataFrobtzz \
    --start-time-filter "{ \"latestDate\" : 1377129600, \"oldestDate\" :
 1370044800 }"
```

출력:

```
{
    "count": 4,
    "truncated": false
}
```

'절단됨'이 `true`이면 'count'는 Amazon 에서 반환할 수 있는 최대 수를 나타냅니다SWF. 추가 결과는 잘립니다.

반환되는 결과 수를 줄이려면 다음을 수행할 수 있습니다.

`--start-time-filter` 값을 수정하여 검색되는 시간 범위를 좁힙니다. `--close-status-filter`, `--execution-filter` `--tag-filter` 또는 `--type-filter` 인수를 사용하여 결과를 추가로 필터링합니다. 각 항목은 상호 배타적입니다. 요청에서 이러한 항목 중 하나만 지정할 수 있습니다.

자세한 내용은 Amazon Simple Workflow Service API 참조 CountOpenWorkflowExecutions 의 섹션을 참조하세요.

- 자세한 API 내용은 명령 참조CountOpenWorkflowExecutions의 섹션을 참조하세요. AWS CLI

### deprecate-domain

다음 코드 예시에서는 deprecate-domain을 사용하는 방법을 보여 줍니다.

AWS CLI

도메인 사용 중단

도메인을 사용 중지하려면(도메인을 여전히 볼 수 있지만 새 워크플로 실행을 생성하거나 유형을 등록할 수 없음) swf deprecate-domain을 사용합니다. 필수 파라미터가 하나 있으며 --name입니다. 이 파라미터는 사용하지 않도록 설정할 도메인의 이름을 가져옵니다.

```
aws swf deprecate-domain \
    --name MyNeatNewDomain ""
```

register-domain과 마찬가지로 출력이 반환되지 않습니다. 그러나 list-domains를 사용하여 등록된 도메인을 보는 경우 도메인이 더 이상 사용되지 않고 반환된 데이터에 더 이상 표시되지 않습니다.

```
aws swf list-domains \
    --registration-status REGISTERED
        {
            "domainInfos": [
                {
                    "status": "REGISTERED",
                    "name": "DataFrobotz"
                },
                {
                    "status": "REGISTERED",
                    "name": "erontest"
                }
            ]
        }
```

--registration-status DEPRECATED 와 함께 를 사용하면 더 이상 사용되지 않는 도메인list-domains이 표시됩니다.

```
aws swf list-domains \
```

```
    --registration-status DEPRECATED
        {
            "domainInfos": [
                {
                    "status": "DEPRECATED",
                    "name": "MyNeatNewDomain"
                }
            ]
        }
```

를 사용하여 더 이상 사용되지 않는 도메인describe-domain에 대한 정보를 가져올 수 있습니다.

```
aws swf describe-domain \
    --name MyNeatNewDomain
        {
            "domainInfo": {
                "status": "DEPRECATED",
                "name": "MyNeatNewDomain"
            },
            "configuration": {
                "workflowExecutionRetentionPeriodInDays": "0"
            }
        }
```

Amazon Simple Workflow Service API 참조의 또한DeprecateDomain 참조

- 자세한 API 내용은 명령 참조DeprecateDomain의 섹션을 참조하세요. AWS CLI

### describe-domain

다음 코드 예시에서는 describe-domain을 사용하는 방법을 보여 줍니다.

AWS CLI

도메인에 대한 정보 가져오기

특정 도메인에 대한 자세한 정보를 얻으려면 swf describe-domain 명령을 사용합니다. 필수 파라미터는 입니다. --name이 파라미터는 정보를 원하는 도메인의 이름을 가져옵니다.

```
aws swf describe-domain \
    --name DataFrobotz
        {
```

```
            "domainInfo": {
                "status": "REGISTERED",
                "name": "DataFrobotz"
            },
            "configuration": {
                "workflowExecutionRetentionPeriodInDays": "1"
            }
        }
```

describe-domain 를 사용하여 더 이상 사용되지 않는 도메인에 대한 정보를 가져올 수도 있습니다.

```
aws swf describe-domain \
    --name MyNeatNewDomain
        {
            "domainInfo": {
                "status": "DEPRECATED",
                "name": "MyNeatNewDomain"
            },
            "configuration": {
                "workflowExecutionRetentionPeriodInDays": "0"
            }
        }
```

Amazon Simple Workflow Service API 참조의 또한DescribeDomain 참조

- 자세한 API 내용은 명령 참조DescribeDomain의 섹션을 참조하세요. AWS CLI

### list-activity-types

다음 코드 예시에서는 list-activity-types을 사용하는 방법을 보여 줍니다.

AWS CLI

활동 유형 나열

도메인의 활동 유형 목록을 가져오려면 를 사용합니다swf list-activity-types. --domain 및 --registration-status 인수가 필요합니다.

```
aws swf list-activity-types \
    --domain DataFrobtzz \
    --registration-status REGISTERED
```

출력:

```
{
    "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"
        }
    ]
}
```

--name 인수를 사용하여 특정 이름의 활동 유형만 선택할 수 있습니다.

```
aws swf list-activity-types \
    --domain DataFrobtzz \
    --registration-status REGISTERED \
    --name "send-subscription-success"
```

출력:

```
{
    "typeInfos": [
        {
            "status": "REGISTERED",
            "creationDate": 1371454150.909,
            "activityType": {
                "version": "1",
                "name": "send-subscription-success"
            },
            "description": "subscribe send-subscription-success activity"
        }
    ]
}
```

페이지에서 결과를 검색하려면 --maximum-page-size 인수를 설정할 수 있습니다. 결과 페이지에 맞는 것보다 더 많은 결과가 반환되면 결과 세트에 “nextPageToken”가 반환됩니다.

```
aws swf list-activity-types \
    --domain DataFrobtzz \
    --registration-status REGISTERED \
    --maximum-page-size 2
```

출력:

```
{
```

```
    "nextPageToken": "AAAAKgAAAAEAAAAAAAAAA1Gp1BelJq
+PmHvAnDxJYbup8+0R4LVtbXLDl7QNY7C3OpHo9Sz06D/GuFz1OyC73umBQ1tOPJ/gC/
aYpzDMqUIWIA1T9W0s2DryyZX4OC/6Lhk9/
o5kdsuWMSBkHhgaZjgwp3WJINIFJFdaSMxY2vYAX7AtRtpcqJuBDDRE9RaRqDGYqIYUMltarkiqpSY1ZVveBasBvlvyU
WGAaqehiDz7/JzLT/wWNNUMOd+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"
        }
    ]
}
```

--next-page-token 인수의 다음 호출list-activity-types에 nextPageToken 값을 전달하여 결과의 다음 페이지를 검색할 수 있습니다.

```
aws swf list-activity-types \
    --domain DataFrobtzz \
    --registration-status REGISTERED \
    --maximum-page-size 2 \
    --next-page-token "AAAAKgAAAAEAAAAAAAAAA1Gp1BelJq
+PmHvAnDxJYbup8+0R4LVtbXLDl7QNY7C3OpHo9Sz06D/GuFz1OyC73umBQ1tOPJ/gC/
aYpzDMqUIWIA1T9W0s2DryyZX4OC/6Lhk9/
o5kdsuWMSBkHhgaZjgwp3WJINIFJFdaSMxY2vYAX7AtRtpcqJuBDDRE9RaRqDGYqIYUMltarkiqpSY1ZVveBasBvlvyU
WGAaqehiDz7/JzLT/wWNNUMOd+Nhe"
```

출력:

```
{
```

```
    "nextPageToken": "AAAAKgAAAAEAAAAAAAAAAw+7LZ4GRZPzTqBHsp2wBxWB8m1sgLCclgCuq3J+h/
m3+vOfFqtkcjLwV5cc4OjNAzTCuq/
XcylPumGwkjbajtqpZpbqOcVNfjFxGoi0LB2Olbvv0krbUISBvlpFPmSWpDSZJsxg5UxCcweteSlFn1PNSZ/
MoinBZo8OTkjMuzcsTuKOzH9wCaR8ITcALJ3SaqHU3pyIRS5hPmFA3OLIc8zaAepjlaujo6hntNSCruB4"
    "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"
        }
    ]
}
```

반환할 결과가 아직 더 있는 경우 결과와 함께 “nextPageToken”가 반환됩니다. 반환할 결과 페이지가 더 이상 없는 경우 결과 세트에 “nextPageToken”가 반환되지 않습니다.

`--reverse-order` 인수를 사용하여 반환된 결과의 순서를 반대로 할 수 있습니다. 이는 호출된 결과에도 영향을 미칩니다.

```
aws swf list-activity-types \
    --domain DataFrobtzz \
    --registration-status REGISTERED \
    --maximum-page-size 2 \
    --reverse-order
```

출력:

```
{
    "nextPageToken": "AAAAKgAAAAEAAAAAAAAAwXcpu5ePSyQkrC
+8WMbmSrenuZC2ZkIXQYBPB/b9xIOVkj+bMEFhGj0KmmJ4rF7iddhjf7UMYCsfGkEn7mk
```

```
+yMCgVc1JxDWmB0EH46bhcmcLmYNQihMDmUWocpr7To6/R7CLu0St1gkFayxOidJXErQW0zdNfQaIWAnF/
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"
        }
    ]
}
```

Amazon Simple Workflow Service API 참조의 또한ListActivityTypes 참조

- 자세한 API 내용은 명령 참조ListActivityTypes의 섹션을 참조하세요. AWS CLI

### `list-domains`

다음 코드 예시에서는 `list-domains`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 등록된 도메인을 나열하려면

다음 `list-domains` 명령 예제에서는 계정에 등록한 `REGISTERED` SWF 도메인을 나열합니다.

```
aws swf list-domains \
    --registration-status REGISTERED
```

출력:

```
{
  "domainInfos": [
    {
      "status": "REGISTERED",
      "name": "DataFrobotz"
    },
    {
      "status": "REGISTERED",
      "name": "erontest"
    }
  ]
}
```

자세한 내용은 Amazon Simple Workflow Service API 참조ListDomains의 섹션을 참조하세요.

예제 2: 더 이상 사용되지 않는 도메인을 나열하려면

다음 `list-domains` 명령 예제에서는 계정에 등록한 DEPRECATED SWF 도메인을 나열합니다. 더 이상 사용되지 않는 도메인은 새 워크플로 또는 활동을 등록할 수 없지만 여전히 쿼리할 수 있는 도메인입니다.

```
aws swf list-domains \
    --registration-status DEPRECATED
```

출력:

```
{
  "domainInfos": [
    {
      "status": "DEPRECATED",
      "name": "MyNeatNewDomain"
    }
  ]
}
```

자세한 내용은 Amazon Simple Workflow Service API 참조ListDomains의 섹션을 참조하세요.

예제 3: 등록된 도메인의 첫 페이지를 나열하려면

다음 `list-domains` 명령 예제에서는 `--maximum-page-size` 옵션을 사용하여 계정에 등록한 첫 번째 페이지 REGISTERED SWF 도메인을 나열합니다.

```
aws swf list-domains \
    --registration-status REGISTERED \
    --maximum-page-size 1
```

출력:

```
{
    "domainInfos": [
        {
            "status": "REGISTERED",
            "name": "DataFrobotz"
        }
    ],
"nextPageToken": "AAAAKgAAAAEAAAAAAAAAA2QJKNtidVgd49TTeNwYcpD
+QKT2ynuEbibcQWe2QKrslMGe63gpS0MgZGpcpoKttL4OCXRFn98Xif557it
+wSZUsvUDtImjDLvguyuyyFdIZtvIxIKEOPm3k2r4OjAGaFsGOuVbrKljvla7wdU7FYH3OlkNCP8b7PBj9SBkUyGoiAg
}
```

자세한 내용은 Amazon Simple Workflow Service API 참조ListDomains의 섹션을 참조하세요.

예제 4: 등록된 도메인의 지정된 단일 페이지를 나열하려면

다음 `list-domains` 명령 예제에서는 `--maximum-page-size` 옵션을 사용하여 계정에 등록한 첫 번째 페이지 REGISTERED SWF 도메인을 나열합니다.

다시 호출하면 이번에는 `--next-page-token` 인수nextPageToken에서 값을 제공하면서 또 다른 결과 페이지가 표시됩니다.

```
aws swf list-domains \
    --registration-status REGISTERED \
    --maximum-page-size 1 \
    --next-page-token "AAAAKgAAAAEAAAAAAAAAA2QJKNtidVgd49TTeNwYcpD
+QKT2ynuEbibcQWe2QKrslMGe63gpS0MgZGpcpoKttL4OCXRFn98Xif557it
+wSZUsvUDtImjDLvguyuyyFdIZtvIxIKEOPm3k2r4OjAGaFsGOuVbrKljvla7wdU7FYH3OlkNCP8b7PBj9SBkUyGoiAg
```

출력:

```
{
    "domainInfos": [
        {
            "status": "REGISTERED",
```

```
            "name": "erontest"
        }
    ]
}
```

더 이상 가져올 결과 페이지가 없으면 nextPageToken이 결과에 반환됩니다.

자세한 내용은 Amazon Simple Workflow Service API 참조ListDomains의 섹션을 참조하세요.

- 자세한 API 내용은 명령 참조ListDomains의 섹션을 참조하세요. AWS CLI

### list-workflow-types

다음 코드 예시에서는 list-workflow-types을 사용하는 방법을 보여 줍니다.

AWS CLI

워크플로 유형 나열

도메인의 워크플로 유형 목록을 가져오려면 를 사용합니다swf list-workflow-types. --domain 및 --registration-status 인수가 필요합니다. 다음은 간단한 예입니다.

```
aws swf list-workflow-types \
    --domain DataFrobtzz \
    --registration-status REGISTERED
```

출력:

```
{
    "typeInfos": [
        {
            "status": "REGISTERED",
            "creationDate": 1371454149.598,
            "description": "DataFrobtzz subscribe workflow",
            "workflowType": {
                "version": "v3",
                "name": "subscribe"
            }
        }
    ]
}
```

와 마찬가지로 --name 인수를 사용하여 특정 이름의 워크플로 유형만 선택하고 와 협력하여 --maximum-page-size 인수를 사용하여 결과를 호출--next-page-token할 list-activity-types수 있습니다. 결과가 반환되는 순서를 반대로 하려면 를 사용합니다--reverse-order.

Amazon Simple Workflow Service API 참조의 또한ListWorkflowTypes 참조

- 자세한 API 내용은 명령 참조ListWorkflowTypes의 섹션을 참조하세요. AWS CLI

### register-domain

다음 코드 예시에서는 register-domain을 사용하는 방법을 보여 줍니다.

AWS CLI

도메인 등록

를 AWS CLI 사용하여 새 도메인을 등록할 수 있습니다. swf register-domain 명령을 사용합니다. 두 가지 필수 파라미터--name, 즉 는 도메인 이름을, --workflow-execution-retention-period-in-days는 정수를 사용하여 이 도메인에서 워크플로 실행 데이터를 보존하는 일수를 최대 90일까지 지정합니다(자세한 내용은 SWF FAQ <https://aws.amazon.com/swf/faqs/#retain_limit> 참조). 지정된 일수가 경과한 후에는 워크플로 실행 데이터가 유지되지 않습니다.

```
aws swf register-domain \
    --name MyNeatNewDomain \
    --workflow-execution-retention-period-in-days 0
    ""
```

도메인을 등록하면 아무것도 반환되지 않지만(“”) swf list-domains 또는 를 사용하여 새 도메인을 swf describe-domain 볼 수 있습니다.

```
aws swf list-domains \
    --registration-status REGISTERED
        {
            "domainInfos": [
                {
                    "status": "REGISTERED",
                    "name": "DataFrobotz"
                },
                {
                    "status": "REGISTERED",
                    "name": "MyNeatNewDomain"
```

```
            },
            {
                "status": "REGISTERED",
                "name": "erontest"
            }
        ]
    }
```

`swf describe-domain` 사용:

```
aws swf describe-domain --
name MyNeatNewDomain
{
    "domainInfo": {
        "status": "REGISTERED",
        "name": "MyNeatNewDomain"
    },
    "configuration": {
        "workflowExecutionRetentionPeriodInDays": "0"

  }
}
```

Amazon Simple Workflow Service API 참조의 또한RegisterDomain 참조

- 자세한 API 내용은 명령 참조RegisterDomain의 섹션을 참조하세요. AWS CLI

### `register-workflow-type`

다음 코드 예시에서는 `register-workflow-type`을 사용하는 방법을 보여 줍니다.

AWS CLI

워크플로 유형 등록

워크플로 유형을 에 등록하려면 `swf register-workflow-type` 명령을 AWS CLI사용합니다.

```
aws swf register-workflow-type \
    --domain DataFrobtzz \
    --name "MySimpleWorkflow" \
    --workflow-version "v1"
```

성공하면 명령이 출력을 생성하지 않습니다.

오류(예: 동일한 워크플로를 두 번 등록하거나 존재하지 않는 도메인을 지정하려고 하는 경우)가 발생하면 에서 응답을 받게 됩니다JSON.

```
{
    "message": "WorkflowType=[name=MySimpleWorkflow, version=v1]",
    "__type": "com.amazonaws.swf.base.model#TypeAlreadyExistsFault"
}
```

`--domain`, `--name` 및 는 `--workflow-version` 필수입니다. 워크플로 설명, 제한 시간 및 하위 워크플로 정책을 설정할 수도 있습니다.

자세한 내용은 Amazon Simple Workflow Service API 참조RegisterWorkflowType의 섹션을 참조하세요.

- 자세한 API 내용은 명령 참조RegisterWorkflowType의 섹션을 참조하세요. AWS CLI

## 를 사용한 Systems Manager 예제 AWS CLI

다음 코드 예제에서는 Systems Manager AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `add-tags-to-resource`

다음 코드 예시에서는 `add-tags-to-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 유지 관리 기간에 태그를 추가하는 방법

다음 add-tags-to-resource 예제에서는 지정된 유지 관리 기간에 태그를 추가합니다.

```
aws ssm add-tags-to-resource \
    --resource-type "MaintenanceWindow" \
    --resource-id "mw-03eb9db428EXAMPLE" \
    --tags "Key=Stack,Value=Production"
```

이 명령은 출력을 생성하지 않습니다.

예제 2: 파라미터에 태그를 추가하는 방법

다음 add-tags-to-resource 예제에서는 지정된 파라미터에 두 개의 태그를 추가합니다.

```
aws ssm add-tags-to-resource \
    --resource-type "Parameter" \
    --resource-id "My-Parameter" \
    --tags '[{"Key":"Region","Value":"East"},{"Key":"Environment",
 "Value":"Production"}]'
```

이 명령은 출력을 생성하지 않습니다.

예제 3: SSM 문서에 태그를 추가하려면

다음 add-tags-to-resource 예제에서는 지정된 문서에 태그를 추가합니다.

```
aws ssm add-tags-to-resource \
    --resource-type "Document" \
    --resource-id "My-Document" \
    --tags "Key=Quarter,Value=Q322"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Systems Manager 사용 설명서의 Systems Manager 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조AddTagsToResource의 섹션을 참조하세요. AWS CLI

### associate-ops-item-related-item

다음 코드 예시에서는 associate-ops-item-related-item을 사용하는 방법을 보여 줍니다.

AWS CLI

관련 항목을 연결하려면

다음 `associate-ops-item-related-item` 예제는 관련 항목을 에 연결합니다 OpsItem.

```
aws ssm associate-ops-item-related-item \
    --ops-item-id "oi-649fExample" \
    --association-type "RelatesTo" \
    --resource-type "AWS::SSMIncidents::IncidentRecord" \
    --resource-uri "arn:aws:ssm-incidents::111122223333:incident-record/Example-Response-Plan/c2bde883-f7d5-343a-b13a-bf5fe9ea689f"
```

출력:

```
{
    "AssociationId": "61d7178d-a30d-4bc5-9b4e-a9e74EXAMPLE"
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 에서 Incident Manager 인시던트 작업을 OpsCenter 참조하세요.

- 자세한 API 내용은 명령 참조AssociateOpsItemRelatedItem의 섹션을 참조하세요. AWS CLI

### `cancel-command`

다음 코드 예시에서는 `cancel-command`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 모든 인스턴스에 대한 명령을 취소하는 방법

다음 `cancel-command` 예제에서는 모든 인스턴스에 대해 이미 실행 중인 지정된 명령을 취소하려고 시도합니다.

```
aws ssm cancel-command \
    --command-id "662add3d-5831-4a10-b64a-f2ff3EXAMPLE"
```

이 명령은 출력을 생성하지 않습니다.

예제 2: 특정 인스턴스의 명령을 취소하는 방법

다음 `cancel-command` 예제에서는 지정된 인스턴스에 대한 명령만 취소하려고 시도합니다.

```
aws ssm cancel-command \
    --command-id "662add3d-5831-4a10-b64a-f2ff3EXAMPLE"
    --instance-ids "i-02573cafcfEXAMPLE"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Systems Manager 사용 설명서의 Systems Manager 파라미터 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조CancelCommand의 섹션을 참조하세요. AWS CLI

### cancel-maintenance-window-execution

다음 코드 예시에서는 cancel-maintenance-window-execution을 사용하는 방법을 보여 줍니다.

AWS CLI

유지 관리 기간 실행을 취소하려면

이 cancel-maintenance-window-execution 예제는 이미 진행 중인 지정된 유지 관리 기간 실행을 중지합니다.

```
aws ssm cancel-maintenance-window-execution \
    --window-execution-id j2l8d5b5c-mw66-tk4d-r3g9-1d4d1EXAMPLE
```

출력:

```
{
    "WindowExecutionId": "j2l8d5b5c-mw66-tk4d-r3g9-1d4d1EXAMPLE"
}
```

자세한 내용은 Systems Manager 사용 설명서의 Systems Manager Maintenance Windows 자습서 (AWS CLI)를 참조하세요. AWS

- 자세한 API 내용은 명령 참조CancelMaintenanceWindowExecution의 섹션을 참조하세요. AWS CLI

### create-activation

다음 코드 예시에서는 create-activation을 사용하는 방법을 보여 줍니다.

AWS CLI

관리형 인스턴스 활성화를 생성하는 방법

다음 create-activation 예제에서는 관리형 인스턴스 활성화를 생성합니다.

```
aws ssm create-activation \
    --default-instance-name "HybridWebServers" \
    --iam-role "HybridWebServersRole" \
    --registration-limit 5
```

출력:

```
{
    "ActivationId": "5743558d-563b-4457-8682-d16c3EXAMPLE",
    "ActivationCode": "dRmgnYaFv567vEXAMPLE"
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 4단계: 하이브리드 환경을 위한 관리형 인스턴스 활성화 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateActivation의 섹션을 참조하세요. AWS CLI

### create-association-batch

다음 코드 예시에서는 create-association-batch을 사용하는 방법을 보여 줍니다.

AWS CLI

다중 연결을 생성하는 방법

이 예제에서는 구성 문서를 여러 인스턴스와 연결합니다. 출력은 해당하는 경우 성공한 작업과 실패한 작업의 목록을 반환합니다.

명령:

```
aws ssm create-association-batch --entries "Name=AWS-
UpdateSSMAgent,InstanceId=i-1234567890abcdef0" "Name=AWS-
UpdateSSMAgent,InstanceId=i-9876543210abcdef0"
```

출력:

```
{
  "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": []
}
```

- 자세한 API 내용은 명령 참조CreateAssociationBatch의 섹션을 참조하세요. AWS CLI

### `create-association`

다음 코드 예시에서는 `create-association`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 인스턴스를 사용하여 문서를 연결하려면 IDs

이 예제에서는 인스턴스 를 사용하여 구성 문서를 인스턴스와 연결합니다IDs.

```
aws ssm create-association \
    --instance-id "i-0cb2b964d3e14fd9f" \
    --name "AWS-UpdateSSMAgent"
```

출력:

```
{
    "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"
            }
        ]
    }
}
```

자세한 내용은 Systems Manager 참조CreateAssociation의 섹션을 참조하세요. AWS API

예제 2: 대상을 사용하여 문서를 연결하는 방법

이 예제에서는 대상을 사용하여 구성 문서를 인스턴스와 연결합니다.

```
aws ssm create-association \
    --name "AWS-UpdateSSMAgent" \
    --targets "Key=instanceids,Values=i-0cb2b964d3e14fd9f"
```

출력:

```
{
    "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"
            }
        ]
    }
}
```

자세한 내용은 Systems Manager 참조CreateAssociation의 섹션을 참조하세요. AWS API

예제 3: 한 번만 실행되는 연결을 생성하는 방법

이 예제에서는 지정된 날짜 및 시간에 한 번만 실행되는 새 연결을 생성합니다. 과거 또는 현재 날짜(처리 시점을 기준으로 해당 날짜가 과거임)에 생성된 연결은 즉시 실행됩니다.

```
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
```

출력:

```
{
    "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"
            }
        ]
    }
}
```

자세한 내용은 Systems Manager AWS 사용 설명서CreateAssociation의 Systems Manager 참조 또는 참조: Systems Manager용 Cron 및 rate 표현식의 섹션을 참조하세요. AWS API

- 자세한 API 내용은 명령 참조CreateAssociation의 섹션을 참조하세요. AWS CLI

### `create-document`

다음 코드 예시에서는 `create-document`을 사용하는 방법을 보여 줍니다.

AWS CLI

문서를 생성하는 방법

다음 `create-document` 예제에서는 Systems Manager 문서를 생성합니다.

```
aws ssm create-document \
    --content file://exampleDocument.yml \
    --name "Example" \
    --document-type "Automation" \
    --document-format YAML
```

출력:

```
{
    "DocumentDescription": {
        "Hash": "fc2410281f40779e694a8b95975d0f9f316da8a153daa94e3d9921102EXAMPLE",
```

```
        "HashType": "Sha256",
        "Name": "Example",
        "Owner": "29884EXAMPLE",
        "CreatedDate": 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": []
    }
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 Systems Manager 문서 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreateDocument의 섹션을 참조하세요. AWS CLI

### `create-maintenance-window`

다음 코드 예시에서는 `create-maintenance-window`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 유지 관리 기간을 생성하는 방법

다음 create-maintenance-window 예제에서는 필요한 경우 5분마다 최대 2시간 동안 유지 관리 기간 실행 종료 1시간 이내에 새 작업 시작을 방지하는 새 유지 관리 기간을 생성하고, 연결되지 않은 대상(유지 관리 기간에 등록되지 않은 인스턴스)을 허용하며, 생성자가 자습서에서 사용하려는 사용자 지정 태그 사용을 통해 이를 나타냅니다.

```
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"
```

출력:

```
{
    "WindowId": "mw-0c50858d01EXAMPLE"
}
```

예제 2: 한 번만 실행되는 유지 관리 기간을 생성하는 방법

다음 create-maintenance-window 예제에서는 지정된 날짜 및 시간에 한 번만 실행되는 새 유지 관리 기간을 생성합니다.

```
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"
```

출력:

```
{
    "WindowId": "mw-01234567890abcdef"
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 유지 관리 기간을 참조하세요.

- 자세한 API 내용은 명령 참조CreateMaintenanceWindow의 섹션을 참조하세요. AWS CLI

### `create-ops-item`

다음 코드 예시에서는 `create-ops-item`을 사용하는 방법을 보여 줍니다.

AWS CLI

를 생성하려면 OpsItems

다음 `create-ops-item` 예제에서는 의 the /aws/resources 키를 사용하여 Amazon DynamoDB 관련 리소스를 OpsItem 사용하여 를 OperationalData 생성합니다.

```
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"
```

출력:

```
{
    "OpsItemId": "oi-1a2b3c4d5e6f"
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 생성을 OpsItems 참조하세요.

- 자세한 API 내용은 명령 참조CreateOpsItem의 섹션을 참조하세요. AWS CLI

### `create-patch-baseline`

다음 코드 예시에서는 `create-patch-baseline`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 자동 승인을 사용하여 패치 기준을 생성하는 방법

다음 create-patch-baseline 예제에서는 Microsoft에서 릴리스하고 7일 후에 프로덕션 환경에 대한 패치를 승인하는 Windows Server용 패치 기준을 생성합니다.

```
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"
```

출력:

```
{
    "BaselineId": "pb-045f10b4f3EXAMPLE"
}
```

예제 2: 승인 마감일이 포함된 패치 기준을 생성하는 방법

다음 create-patch-baseline 예제에서는 2020년 7월 7일을 포함하여 해당 날짜 이전에 릴리스된 프로덕션 환경에 대한 패치를 승인하는 Windows Server용 패치 기준을 생성합니다.

```
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"
```

출력:

```
{
    "BaselineId": "pb-045f10b4f3EXAMPLE"
}
```

예제 3: JSON 파일에 저장된 승인 규칙으로 패치 기준을 생성하려면

다음 create-patch-baseline 예제에서는 Amazon Linux 2017.09용 패치 기준을 생성합니다. 여기에서는 릴리스하고 7일 후에 프로덕션 환경에 대한 패치를 승인하고 패치 기준에 대한 승인 규칙을 지정하며 패치에 대한 사용자 지정 리포지토리를 지정합니다.

```
aws ssm create-patch-baseline \
    --cli-input-json file://my-amazon-linux-approval-rules-and-repo.json
```

my-amazon-linux-approval-rules-and-repo.json의 콘텐츠:

```
{
    "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"
        }
    ]
}
```

예제 4: 승인된 패치와 거부된 패치를 지정하는 패치 기준을 생성하는 방법

다음 create-patch-baseline 예제에서는 기본 승인 규칙의 예외로 승인 및 거부할 패치를 명시적으로 지정합니다.

```
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"
```

자세한 내용은 AWS Systems Manager 사용 설명서의 사용자 지정 패치 기준 생성을 참조하세요.

- 자세한 API 내용은 명령 참조CreatePatchBaseline의 섹션을 참조하세요. AWS CLI

### create-resource-data-sync

다음 코드 예시에서는 create-resource-data-sync을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 데이터 동기화를 생성하려면

이 예제에서는 리소스 데이터 동기화를 생성합니다. 명령이 성공해도 출력은 없습니다.

명령:

```
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"
```

- 자세한 API 내용은 명령 참조CreateResourceDataSync의 섹션을 참조하세요. AWS CLI

### delete-activation

다음 코드 예시에서는 delete-activation을 사용하는 방법을 보여 줍니다.

AWS CLI

관리형 인스턴스 활성화를 삭제하는 방법

다음 delete-activation 예제에서는 관리형 인스턴스 활성화를 삭제합니다.

```
aws ssm delete-activation \
    --activation-id "aa673477-d926-42c1-8757-1358cEXAMPLE"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Systems Manager 사용 설명서의 하이브리드 환경을 위한 Systems Manager 설정을 참조하세요. AWS

- 자세한 API 내용은 명령 참조DeleteActivation의 섹션을 참조하세요. AWS CLI

### delete-association

다음 코드 예시에서는 delete-association을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 연결 ID를 사용하여 연결을 삭제하는 방법

다음 delete-association 예제에서는 지정된 연결 ID의 연결을 삭제합니다. 명령이 성공해도 출력은 없습니다.

```
aws ssm delete-association \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Systems Manager 사용 설명서의 새 연결 버전 편집 및 생성을 참조하세요.

예제 2: 연결을 삭제하는 방법

다음 delete-association 예제에서는 인스턴스와 문서 간 연결을 삭제합니다. 명령이 성공해도 출력은 없습니다.

```
aws ssm delete-association \
    --instance-id "i-1234567890abcdef0" \
    --name "AWS-UpdateSSMAgent"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Systems Manager 사용 설명서의 Systems Manager에서 연결 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteAssociation의 섹션을 참조하세요. AWS CLI

## delete-document

다음 코드 예시에서는 delete-document을 사용하는 방법을 보여 줍니다.

AWS CLI

문서를 삭제하는 방법

다음 delete-document 예제에서는 Systems Manager 문서를 삭제합니다.

```
aws ssm delete-document \
    --name "Example"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Systems Manager 사용 설명서의 Systems Manager 문서 생성을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteDocument의 섹션을 참조하세요. AWS CLI

### `delete-inventory`

다음 코드 예시에서는 `delete-inventory`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 인벤토리 유형을 삭제하려면

이 예제에서는 사용자 지정 인벤토리 스키마를 삭제합니다.

명령:

```
aws ssm delete-inventory --type-name "Custom:RackInfo" --schema-delete-
option "DeleteSchema"
```

출력:

```
{
  "DeletionId": "d72ac9e8-1f60-4d40-b1c6-bf8c78c68c4d",
  "TypeName": "Custom:RackInfo",
  "DeletionSummary": {
      "TotalCount": 1,
      "RemainingCount": 1,
      "SummaryItems": [
          {
              "Version": "1.0",
              "Count": 1,
              "RemainingCount": 1
          }
      ]
  }
}
```

사용자 지정 인벤토리 유형을 비활성화하려면

이 예제에서는 사용자 지정 인벤토리 스키마를 비활성화합니다.

명령:

```
aws ssm delete-inventory --type-name "Custom:RackInfo" --schema-delete-
option "DisableSchema"
```

출력:

```
{
  "DeletionId": "6961492a-8163-44ec-aa1e-923364dd0850",
  "TypeName": "Custom:RackInformation",
  "DeletionSummary": {
      "TotalCount": 0,
      "RemainingCount": 0,
      "SummaryItems": []
  }
}
```

- 자세한 API 내용은 명령 참조DeleteInventory의 섹션을 참조하세요. AWS CLI

### `delete-maintenance-window`

다음 코드 예시에서는 `delete-maintenance-window`을 사용하는 방법을 보여 줍니다.

AWS CLI

유지 관리 기간을 삭제하는 방법

이 `delete-maintenance-window` 예제에서는 지정된 유지 관리 기간을 제거합니다.

```
aws ssm delete-maintenance-window \
    --window-id "mw-1a2b3c4d5e6f7g8h9"
```

출력:

```
{
    "WindowId":"mw-1a2b3c4d5e6f7g8h9"
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 유지 관리 기간 삭제(AWS CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteMaintenanceWindow의 섹션을 참조하세요. AWS CLI

### delete-parameter

다음 코드 예시에서는 delete-parameter을 사용하는 방법을 보여 줍니다.

AWS CLI

파라미터를 삭제하는 방법

다음 delete-parameter 예제에서는 지정된 단일 파라미터를 삭제합니다.

```
aws ssm delete-parameter \
    --name "MyParameter"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS  Systems Manager 사용 설명서의 Parameter Store 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteParameter의 섹션을 참조하세요. AWS CLI

### delete-parameters

다음 코드 예시에서는 delete-parameters을 사용하는 방법을 보여 줍니다.

AWS CLI

파라미터 목록을 삭제하려면

다음 delete-parameters 예제에서는 지정된 파라미터를 삭제합니다.

```
aws ssm delete-parameters \
    --names "MyFirstParameter" "MySecondParameter" "MyInvalidParameterName"
```

출력:

```
{
    "DeletedParameters": [
        "MyFirstParameter",
        "MySecondParameter"
    ],
    "InvalidParameters": [
        "MyInvalidParameterName"
    ]
}
```

자세한 내용은 AWS  Systems Manager 사용 설명서의 Parameter Store 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteParameters의 섹션을 참조하세요. AWS CLI

### delete-patch-baseline

다음 코드 예시에서는 delete-patch-baseline을 사용하는 방법을 보여 줍니다.

AWS CLI

패치 기준을 삭제하는 방법

다음 delete-patch-baseline 예제에서는 지정된 패치 기준을 삭제합니다.

```
aws ssm delete-patch-baseline \
    --baseline-id "pb-045f10b4f382baeda"
```

출력:

```
{
    "BaselineId": "pb-045f10b4f382baeda"
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 패치 기준 업데이트 또는 삭제(콘솔)를 참조하세요.

- 자세한 API 내용은 명령 참조DeletePatchBaseline의 섹션을 참조하세요. AWS CLI

### delete-resource-data-sync

다음 코드 예시에서는 delete-resource-data-sync을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 데이터 동기화를 삭제하려면

이 예제에서는 리소스 데이터 동기화를 삭제합니다. 명령이 성공해도 출력은 없습니다.

명령:

```
aws ssm delete-resource-data-sync --sync-name "ssm-resource-data-sync"
```

- 자세한 API 내용은 명령 참조DeleteResourceDataSync의 섹션을 참조하세요. AWS CLI

### deregister-managed-instance

다음 코드 예시에서는 deregister-managed-instance을 사용하는 방법을 보여 줍니다.

AWS CLI

관리형 인스턴스를 등록 취소하는 방법

다음 deregister-managed-instance 예제에서는 지정된 관리형 인스턴스를 등록 취소합니다.

```
aws ssm deregister-managed-instance
    --instance-id "mi-08ab247cdfEXAMPLE"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Systems Manager 사용 설명서의 하이브리드 환경에서 관리형 인스턴스 등록 취소를 참조하세요.

- 자세한 API 내용은 명령 참조DeregisterManagedInstance의 섹션을 참조하세요. AWS CLI

### deregister-patch-baseline-for-patch-group

다음 코드 예시에서는 deregister-patch-baseline-for-patch-group을 사용하는 방법을 보여 줍니다.

AWS CLI

패치 기준에서 패치 그룹을 등록 취소하는 방법

다음 deregister-patch-baseline-for-patch-group 예제에서는 지정된 패치 기준에서 지정된 패치 그룹을 등록 취소합니다.

```
aws ssm deregister-patch-baseline-for-patch-group \
    --patch-group "Production" \
    --baseline-id "pb-0ca44a362fEXAMPLE"
```

출력:

```
{
  "PatchGroup":"Production",
  "BaselineId":"pb-0ca44a362fEXAMPLE"
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 패치 기준에 패치 그룹 추가를 참조하세요.

- 자세한 API 내용은 명령 참조DeregisterPatchBaselineForPatchGroup의 섹션을 참조하세요. AWS CLI

### deregister-target-from-maintenance-window

다음 코드 예시에서는 deregister-target-from-maintenance-window을 사용하는 방법을 보여 줍니다.

AWS CLI

유지 관리 기간에서 대상을 제거하는 방법

다음 deregister-target-from-maintenance-window 예제에서는 지정된 유지 관리 기간에서 지정된 대상을 제거합니다.

```
aws ssm deregister-target-from-maintenance-window \
    --window-id "mw-ab12cd34ef56gh78" \
    --window-target-id "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
```

출력:

```
{
    "WindowId":"mw-ab12cd34ef56gh78",
    "WindowTargetId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 유지 관리 기간 업데이트(AWS CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조DeregisterTargetFromMaintenanceWindow의 섹션을 참조하세요. AWS CLI

### deregister-task-from-maintenance-window

다음 코드 예시에서는 deregister-task-from-maintenance-window을 사용하는 방법을 보여 줍니다.

AWS CLI

유지 관리 기간에서 작업을 제거하는 방법

다음 deregister-task-from-maintenance-window 예제에서는 지정된 유지 관리 기간에서 지정된 작업을 제거합니다.

```
aws ssm deregister-task-from-maintenance-window \
    --window-id "mw-ab12cd34ef56gh78" \
    --window-task-id "1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d5e6c"
```

출력:

```
{
    "WindowTaskId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d5e6c",
    "WindowId":"mw-ab12cd34ef56gh78"
}
```

자세한 내용은 Systems Manager 사용 설명서의 Systems Manager 유지 관리 Windows 자습서 (AWS CLI)를 참조하세요. AWS

- 자세한 API 내용은 명령 참조DeregisterTaskFromMaintenanceWindow의 섹션을 참조하세요. AWS CLI

### describe-activations

다음 코드 예시에서는 describe-activations을 사용하는 방법을 보여 줍니다.

AWS CLI

활성화를 설명하는 방법

다음 describe-activations 예제에서는 AWS 계정의 활성화에 대한 세부 정보를 나열합니다.

```
aws ssm describe-activations
```

출력:

```
{
    "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,
            "CreatedDate": 1578064132.002
        },
    ]
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 4단계: 하이브리드 환경을 위한 관리형 인스턴스 활성화 생성을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeActivations의 섹션을 참조하세요. AWS CLI

### describe-association-execution-targets

다음 코드 예시에서는 describe-association-execution-targets을 사용하는 방법을 보여 줍니다.

AWS CLI

연결 실행의 세부 정보를 가져오는 방법

다음 describe-association-execution-targets 예제에서는 지정된 연결 실행을 설명합니다.

```
aws ssm describe-association-execution-targets \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab" \
    --execution-id "7abb6378-a4a5-4f10-8312-0123456789ab"
```

출력:

```
{
    "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"
            }
        }
    ]
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 연결 기록 보기를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeAssociationExecutionTargets의 섹션을 참조하세요. AWS CLI

### describe-association-executions

다음 코드 예시에서는 describe-association-executions을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 연결에 대한 모든 실행 세부 정보를 가져오는 방법

다음 describe-association-executions 예제에서는 지정된 연결의 모든 실행을 설명합니다.

```
aws ssm describe-association-executions \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

출력:

```
{
    "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}"
        },
        ...
    ]
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 연결 기록 보기를 참조하세요.

예제 2: 특정 날짜 및 시간 이후 연결에 대한 모든 실행의 세부 정보를 보는 방법

다음 `describe-association-executions` 예제에서는 지정된 날짜 및 시간 이후 연결의 모든 실행을 설명합니다.

```
aws ssm describe-association-executions \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab" \
    --filters "Key=CreatedTime,Value=2019-02-18T16:00:00Z,Type=GREATER_THAN"
```

출력:

```
{
    "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}"
        },
        ...
    ]
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 연결 기록 보기를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeAssociationExecutions의 섹션을 참조하세요. AWS CLI

### describe-association

다음 코드 예시에서는 describe-association을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 연결 세부 정보를 가져오는 방법

다음 describe-association 예제에서는 지정된 연결 ID의 연결을 설명합니다.

```
aws ssm describe-association \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 새 연결 버전 편집 및 생성을 참조하세요.

예제 2: 특정 인스턴스 및 문서에 대한 연결 세부 정보를 가져오는 방법

다음 `describe-association` 예제에서는 인스턴스와 문서 간 연결을 설명합니다.

```
aws ssm describe-association \
    --instance-id "i-1234567890abcdef0" \
    --name "AWS-UpdateSSMAgent"
```

출력:

```
{
    "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"
            }
        ]
    }
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 새 연결 버전 편집 및 생성을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeAssociation의 섹션을 참조하세요. AWS CLI

### describe-automation-executions

다음 코드 예시에서는 describe-automation-executions을 사용하는 방법을 보여 줍니다.

AWS CLI

자동화 실행을 설명하는 방법

다음 describe-automation-executions 예제에서는 자동화 실행에 대한 세부 정보를 표시합니다.

```
aws ssm describe-automation-executions \
    --filters Key=ExecutionId,Values=73c8eef8-f4ee-4a05-820c-e354fEXAMPLE
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 단순 자동화 워크플로 실행을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeAutomationExecutions의 섹션을 참조하세요. AWS CLI

### describe-automation-step-executions

다음 코드 예시에서는 describe-automation-step-executions을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 자동화 실행의 모든 단계를 설명하는 방법

다음 describe-automation-step-executions 예제에서는 자동화 실행 단계에 대한 세부 정보를 표시합니다.

```
aws ssm describe-automation-step-executions \
    --automation-execution-id 73c8eef8-f4ee-4a05-820c-e354fEXAMPLE
```

출력:

```
{
    "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": {}
        }
    ]
}
```

예제 2: 자동화 실행의 특정 단계를 설명하는 방법

다음 `describe-automation-step-executions` 예제에서는 자동화 실행의 특정 단계에 대한 세부 정보를 표시합니다.

```
aws ssm describe-automation-step-executions \
    --automation-execution-id 73c8eef8-f4ee-4a05-820c-e354fEXAMPLE \
    --filters Key=StepExecutionId,Values=95e70479-cf20-4d80-8018-7e4e2EXAMPLE
```

자세한 내용은 AWS Systems Manager 사용 설명서의 자동화 워크플로 단계별 실행(명령줄)을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeAutomationStepExecutions의 섹션을 참조하세요. AWS CLI

### describe-available-patches

다음 코드 예시에서는 `describe-available-patches`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 패치를 가져오는 방법

다음 `describe-available-patches` 예제에서는 MSRC 심각도가 Critical인 Windows Server 2019에서 사용 가능한 모든 패치에 대한 세부 정보를 검색합니다.

```
aws ssm describe-available-patches \
    --
filters "Key=PRODUCT,Values=WindowsServer2019" "Key=MSRC_SEVERITY,Values=Critical"
```

출력:


```
{
    "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"
        },
        ...
```

```
    ]
}
```

특정 패치의 세부 정보를 가져오는 방법

다음 `describe-available-patches` 예제에서는 지정된 패치에 대한 세부 정보를 검색합니다.

```
aws ssm describe-available-patches \
    --filters "Key=PATCH_ID,Values=KB4480979"
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 Patch Manager 작업 작동 방법을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeAvailablePatches의 섹션을 참조하세요. AWS CLI

### describe-document-permission

다음 코드 예시에서는 describe-document-permission을 사용하는 방법을 보여 줍니다.

AWS CLI

문서 권한을 설명하는 방법

다음 describe-document-permission 예제에서는 공개적으로 공유되는 Systems Manager 문서에 대한 권한 세부 정보를 표시합니다.

```
aws ssm describe-document-permission \
    --name "Example" \
    --permission-type "Share"
```

출력:

```
{
    "AccountIds": [
        "all"
    ],
    "AccountSharingInfoList": [
        {
            "AccountId": "all",
            "SharedDocumentVersion": "$DEFAULT"
        }
    ]
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 Systems Manager 문서 공유를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeDocumentPermission의 섹션을 참조하세요. AWS CLI

### describe-document

다음 코드 예시에서는 describe-document을 사용하는 방법을 보여 줍니다.

AWS CLI

문서 세부 정보를 표시하는 방법

다음 describe-document 예제에서는 AWS 계정의 Systems Manager 문서에 대한 세부 정보를 표시합니다.

```
aws ssm describe-document \
    --name "Example"
```

출력:

```
{
    "Document": {
        "Hash": "fc2410281f40779e694a8b95975d0f9f316da8a153daa94e3d9921102EXAMPLE",
        "HashType": "Sha256",
        "Name": "Example",
        "Owner": "29884EXAMPLE",
        "CreatedDate": 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": []
    }
```

```
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 Systems Manager 문서 생성을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeDocument의 섹션을 참조하세요. AWS CLI

### describe-effective-instance-associations

다음 코드 예시에서는 describe-effective-instance-associations을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스에 대한 유효한 연결의 세부 정보를 가져오는 방법

다음 describe-effective-instance-associations 예제에서는 인스턴스에 대한 유효한 연결의 세부 정보를 검색합니다.

명령:

```
aws ssm describe-effective-instance-associations --instance-id "i-1234567890abcdef0"
```

출력:

```
{
    "Associations": [
        {
            "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
            "InstanceId": "i-1234567890abcdef0",
            "Content": "{\n    \"schemaVersion\": \"1.2\",\n    \"description\":
 \"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
 to install. If not specified, the agent will be updated to the latest version.\",
\n            \"type\": \"String\"\n        },\n        \"allowDowngrade\": {\n
           \"default\": \"false\",\n            \"description\": \"(Optional)
 Allow the Amazon SSM Agent service to be downgraded to an earlier version. If
 set to false, the service can be upgraded to newer versions only (default). If
 set to true, specify the earlier version.\",\n            \"type\": \"String\",
\n            \"allowedValues\": [\n                \"true\",\n
 \"false\"\n            ]\n        }\n    },\n    \"runtimeConfig\": {\n
 \"aws:updateSsmAgent\": {\n            \"properties\": [\n                {\n
              \"agentName\": \"amazon-ssm-agent\",\n                \"source\":
```

```
 \"https://s3.{Region}.amazonaws.com/amazon-ssm-{Region}/ssm-agent-manifest.json\",
\n                \"allowDowngrade\": \"{{ allowDowngrade }}\",\n
 \"targetVersion\": \"{{ version }}\"\n                }\n            ]\n        }\n
    }\n}\n",
            "AssociationVersion": "1"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeEffectiveInstanceAssociations의 섹션을 참조하세요. AWS CLI

### describe-effective-patches-for-patch-baseline

다음 코드 예시에서는 describe-effective-patches-for-patch-baseline을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 사용자 지정 패치 기준에서 정의한 모든 패치를 가져오는 방법

다음 describe-effective-patches-for-patch-baseline 예제에서는 현재 AWS 계정의 사용자 지정 패치 기준선으로 정의된 패치를 반환합니다. 사용자 지정 기준의 경우 --baseline-id에는 ID만 필요합니다.

```
aws ssm describe-effective-patches-for-patch-baseline \
    --baseline-id "pb-08b654cf9b9681f04"
```

출력:

```
{
    "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--"
}
```

예제 2: AWS 관리형 패치 기준선으로 정의된 모든 패치를 가져오려면

다음 `describe-effective-patches-for-patch-baseline` 예제에서는 AWS 관리형 패치 기준선으로 정의된 패치를 반환합니다. AWS 관리형 기준의 경우 에 대한 전체 기준이 ARN 필요합니다. `--baseline-id`

```
aws ssm describe-effective-patches-for-patch-baseline \
    --baseline-id "arn:aws:ssm:us-east-2:733109147000:patchbaseline/
pb-020d361a05defe4ed"
```

샘플 출력은 예 1을 참조하세요.

자세한 내용은 AWS Systems Manager 사용 설명서의 보안 패치 선택 방법을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeEffectivePatchesForPatchBaseline의 섹션을 참조하세요. AWS CLI

### describe-instance-associations-status

다음 코드 예시에서는 `describe-instance-associations-status`을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스 연결 상태를 설명하는 방법

이 예제에서는 인스턴스 연결의 세부 정보를 보여줍니다.

명령:

```
aws ssm describe-instance-associations-status --instance-id "i-1234567890abcdef0"
```

출력:

```
{
```

```
  "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"
      }
  ]
}
```

- 자세한 API 내용은 명령 참조DescribeInstanceAssociationsStatus의 섹션을 참조하세요. AWS CLI

### describe-instance-information

다음 코드 예시에서는 describe-instance-information을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 관리형 인스턴스 정보를 설명하는 방법

다음 describe-instance-information 예제에서는 각 관리형 인스턴스의 세부 정보를 검색합니다.

```
aws ssm describe-instance-information
```

예제 2: 특정 관리형 인스턴스에 대한 정보를 설명하는 방법

다음 `describe-instance-information` 예제에서는 관리형 인스턴스 `i-028ea792daEXAMPLE`의 세부 정보를 보여줍니다.

```
aws ssm describe-instance-information \
    --filters "Key=InstanceIds,Values=i-028ea792daEXAMPLE"
```

예제 3: 특정 태그 키를 사용하는 관리형 인스턴스에 대한 정보를 설명하는 방법

다음 `describe-instance-information` 예제에서는 태그 키 `DEV`가 있는 관리형 인스턴스의 세부 정보를 보여줍니다.

```
aws ssm describe-instance-information \
    --filters "Key=tag-key,Values=DEV"
```

출력:

```
{
    "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
                }
            }
        }
```

```
    ]
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 관리형 인스턴스를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeInstanceInformation의 섹션을 참조하세요. AWS CLI

### describe-instance-patch-states-for-patch-group

다음 코드 예시에서는 describe-instance-patch-states-for-patch-group을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 패치 그룹의 인스턴스 상태를 가져오는 방법

다음 describe-instance-patch-states-for-patch-group 예제에서는 지정된 패치 그룹의 인스턴스당 패치 요약 상태에 대한 세부 정보를 검색합니다.

```
aws ssm describe-instance-patch-states-for-patch-group \
    --patch-group "Production"
```

출력:

```
{
    "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
        }
    ]
}
```

예제 2: 패치가 5개 넘게 누락된 패치 그룹의 인스턴스 상태를 가져오는 방법

다음 `describe-instance-patch-states-for-patch-group` 예제에서는 패치가 5개 넘게 누락된 인스턴스에서 지정된 패치 그룹의 패치 요약 상태에 대한 세부 정보를 검색합니다.

```
aws ssm describe-instance-patch-states-for-patch-group \
    --filters Key=MissingCount,Type=GreaterThan,Values=5 \
    --patch-group "Production"
```

출력:

```
{
```

```
    "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
        }
    ]
}
```

예제 3: 재부팅이 필요한 인스턴스가 10개 미만인 패치 그룹의 인스턴스 상태를 가져오는 방법

다음 describe-instance-patch-states-for-patch-group 예제에서는 리부팅해야 하는 패치가 10개 미만인 인스턴스에서 지정된 패치 그룹의 패치 요약 상태에 대한 세부 정보를 검색합니다.

```
aws ssm describe-instance-patch-states-for-patch-group \
    --filters Key=InstalledPendingRebootCount,Type=LessThan,Values=10 \
    --patch-group "Production"
```

출력:

```
{
    "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": "2021-08-046T11:03:50.590000-07:00",
            "OperationEndTime": "2021-08-06T11:04:21.555000-07:00",
            "Operation": "Scan",
            "RebootOption": "NoReboot",
            "CriticalNonCompliantCount": 0,
            "SecurityNonCompliantCount": 1,
            "OtherNonCompliantCount": 0
        }
    ]
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 패치 규정 준수 상태 값 이해를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeInstancePatchStatesForPatchGroup의 섹션을 참조하세요. AWS CLI

### describe-instance-patch-states

다음 코드 예시에서는 describe-instance-patch-states을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스의 패치 요약 상태를 가져오는 방법

이 describe-instance-patch-states 예제에서는 인스턴스의 패치 요약 상태를 가져옵니다.

```
aws ssm describe-instance-patch-states \
    --instance-ids "i-1234567890abcdef0"
```

출력:

```
{
    "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"
        }
    ]
}
```


자세한 내용은 AWS Systems Manager 사용 설명서의 패치 규정 준수 정보를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeInstancePatchStates의 섹션을 참조하세요. AWS CLI

### describe-instance-patches

다음 코드 예시에서는 describe-instance-patches을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 인스턴스의 패치 상태 세부 정보를 가져오는 방법

다음 describe-instance-patches 예제에서는 지정된 인스턴스의 패치에 대한 세부 정보를 검색합니다.

```
aws ssm describe-instance-patches \
```

```
    --instance-id "i-1234567890abcdef0"
```

출력:

```
{
    "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--"
}
```

예제 2: 인스턴스에서 누락 상태의 패치 목록을 가져오는 방법

다음 `describe-instance-patches` 예제에서는 지정된 인스턴스에서 누락 상태인 패치에 대한 정보를 검색합니다.

```
aws ssm describe-instance-patches \
    --instance-id "i-1234567890abcdef0" \
    --filters Key=State,Values=Missing
```

출력:

```
{
    "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--"
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 패치 규정 준수 상태 정보를 참조하세요.

예제 3: 인스턴스에 InstalledTime 지정된 이후 설치된 패치 목록을 가져오려면

다음 `describe-instance-patches` 예제에서는 `--filters` 및 `--query`의 사용을 조합하여 지정된 인스턴스에 대해 지정된 시간 이후에 설치된 패치에 대한 정보를 검색합니다.

```
aws ssm describe-instance-patches \
    --instance-id "i-1234567890abcdef0" \
    --filters Key=State,Values=Installed \
    --query "Patches[?InstalledTime >= `2023-01-01T16:00:00`]"
```

출력:

```
{
    "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--"
```

```
}
```

- 자세한 API 내용은 명령 참조DescribeInstancePatches의 섹션을 참조하세요. AWS CLI

### describe-inventory-deletions

다음 코드 예시에서는 describe-inventory-deletions을 사용하는 방법을 보여 줍니다.

AWS CLI

인벤토리 삭제를 가져오려면

이 예제에서는 인벤토리 삭제 작업에 대한 세부 정보를 검색합니다.

명령:

```
aws ssm describe-inventory-deletions
```

출력:

```
{
  "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
        }
    ]
}
```

특정 인벤토리 삭제에 대한 세부 정보를 가져오려면

이 예제에서는 특정 인벤토리 삭제 작업에 대한 세부 정보를 검색합니다.

명령:

```
aws ssm describe-inventory-deletions --deletion-id "d72ac9e8-1f60-4d40-
b1c6-987654321c4d"
```

출력:

```
{
    "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
    }
  ]
}
```

- 자세한 API 내용은 명령 참조DescribeInventoryDeletions의 섹션을 참조하세요. AWS CLI

### describe-maintenance-window-execution-task-invocations

다음 코드 예시에서는 describe-maintenance-window-execution-task-invocations을 사용하는 방법을 보여 줍니다.

AWS CLI

유지 관리 기간 작업 실행을 위해 수행된 특정 작업 간접 호출을 가져오는 방법

다음 describe-maintenance-window-execution-task-invocations 예제에서는 지정된 유지 관리 기간 실행의 일부로 실행된 작업에 간접 호출을 나열합니다.

```
aws ssm describe-maintenance-window-execution-task-invocations \
    --window-execution-id "518d5565-5969-4cca-8f0e-da3b2a638355" \
    --task-id "ac0c6ae1-daa3-4a89-832e-d384503b6586"
```

출력:

```
{
    "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"
        }
    ]
```

```
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 태스크 및 태스크 실행에 대한 정보 보기 (AWS CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeMaintenanceWindowExecutionTaskInvocations의 섹션을 참조하세요. AWS CLI

### `describe-maintenance-window-execution-tasks`

다음 코드 예시에서는 `describe-maintenance-window-execution-tasks`을 사용하는 방법을 보여 줍니다.

AWS CLI

유지 관리 기간 실행과 연결된 모든 작업을 나열하는 방법

다음 `ssm describe-maintenance-window-execution-tasks` 예제에서는 지정된 유지 관리 기간 실행과 연결된 작업을 나열합니다.

```
aws ssm describe-maintenance-window-execution-tasks \
    --window-execution-id "518d5565-5969-4cca-8f0e-da3b2EXAMPLE"
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 태스크 및 태스크 실행에 대한 정보 보기 (AWS CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeMaintenanceWindowExecutionTasks의 섹션을 참조하세요. AWS CLI

### describe-maintenance-window-executions

다음 코드 예시에서는 describe-maintenance-window-executions을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 유지 관리 기간의 모든 실행을 나열하는 방법

다음 describe-maintenance-window-executions 예제에서는 지정된 유지 관리 기간의 모든 실행을 나열합니다.

```
aws ssm describe-maintenance-window-executions \
    --window-id "mw-ab12cd34eEXAMPLE"
```

출력:

```
{
    "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"
        }
    ]
}
```

예제 2: 지정된 날짜 이전의 유지 관리 기간에 대한 모든 실행을 나열하는 방법

다음 describe-maintenance-window-executions 예제에서는 지정된 날짜 이전에 지정된 유지 관리 기간의 모든 실행을 나열합니다.

```
aws ssm describe-maintenance-window-executions \
    --window-id "mw-ab12cd34eEXAMPLE" \
    --filters "Key=ExecutedBefore,Values=2021-08-03T00:00:00Z"
```

출력:

```
{
    "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"
    }
    ]
}
```

예제 3: 지정된 날짜 이후 유지 관리 기간에 대한 모든 실행을 나열하는 방법

다음 describe-maintenance-window-executions 예제에서는 지정된 날짜 이후에 지정된 유지 관리 기간의 모든 실행을 나열합니다.

```
aws ssm describe-maintenance-window-executions \
    --window-id "mw-ab12cd34eEXAMPLE" \
    --filters "Key=ExecutedAfter,Values=2021-08-04T00:00:00Z"
```

출력:

```
{
```

```
    "WindowExecutions": [
        {
        "WindowId": "mw-ab12cd34eEXAMPLE",
        "WindowExecutionId": "6027b513-64fe-4cf0-be7d-1191aEXAMPLE",
        "Status": "IN_PROGRESS",
        "StartTime": "2021-08-04T11:00:00.000000-07:00"
        }
    ]
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 태스크 및 태스크 실행(AWS CLI)에 대한 정보 보기를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeMaintenanceWindowExecutions의 섹션을 참조하세요. AWS CLI

### describe-maintenance-window-schedule

다음 코드 예시에서는 describe-maintenance-window-schedule을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 유지 관리 기간에 대한 예정된 실행을 나열하려면

다음 describe-maintenance-window-schedule 예제에서는 지정된 유지 관리 기간에 대해 예정된 모든 실행을 나열합니다.

```
aws ssm describe-maintenance-window-schedule \
    --window-id mw-ab12cd34eEXAMPLE
```

출력:

```
{
    "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"
        },
        ...
    ]
}
```

예제 2: 지정된 날짜 이전의 유지 관리 기간에 대해 예정된 모든 실행을 나열하려면

다음 describe-maintenance-window-schedule 예제에서는 지정된 날짜 이전에 발생하는 지정된 유지 관리 기간에 대해 예정된 모든 실행을 나열합니다.

```
aws ssm describe-maintenance-window-schedule \
    --window-id mw-0ecb1226dd7b2e9a6 \
    --filters "Key=ScheduledBefore,Values=2020-02-15T06:00:00Z"
```

예제 3: 지정된 날짜 이후의 유지 관리 기간에 대해 예정된 모든 실행을 나열하려면

다음 describe-maintenance-window-schedule 예제에서는 지정된 날짜 이후에 발생하는 지정된 유지 관리 기간에 대해 예정된 모든 실행을 나열합니다.

```
aws ssm describe-maintenance-window-schedule \
    --window-id mw-0ecb1226dd7b2e9a6 \
    --filters "Key=ScheduledAfter,Values=2020-02-15T06:00:00Z"
```

자세한 내용은 AWS Systems Manager 사용 설명서의 유지 관리 Windows(AWS CLI)에 대한 정보 보기를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeMaintenanceWindowSchedule의 섹션을 참조하세요. AWS CLI

### describe-maintenance-window-targets

다음 코드 예시에서는 describe-maintenance-window-targets을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 유지 관리 기간의 모든 대상을 나열하는 방법

다음 describe-maintenance-window-targets 예제에서는 유지 관리 기간의 모든 대상을 나열합니다.

```
aws ssm describe-maintenance-window-targets \
    --window-id "mw-06cf17cbefEXAMPLE"
```

출력:

```
{
    "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"
        }
    ]
}
```

예제 2: 특정 소유자 정보 값과 일치하는 유지 관리 기간의 대상을 나열하는 방법

이 describe-maintenance-window-targets 예제에서는 특정 값이 있는 유지 관리 기간의 모든 대상을 나열합니다.

```
aws ssm describe-maintenance-window-targets \
    --window-id "mw-0ecb1226ddEXAMPLE" \
    --filters "Key=OwnerInformation,Values=CostCenter1"
```

출력:

```
{
    "Targets": [
        {
            "WindowId": "mw-0ecb1226ddEXAMPLE",
            "WindowTargetId": "da89dcc3-7f9c-481d-ba2b-edcb7d0057f9",
            "ResourceType": "INSTANCE",
            "Targets": [
                {
                    "Key": "tag:Environment",
                    "Values": [
                        "Prod"
                    ]
                }
            ],
            "OwnerInformation": "CostCenter1",
            "Name": "ProdTarget1"
        }
    ]
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 유지 관리 Windows(AWS CLI)에 대한 정보 보기를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeMaintenanceWindowTargets의 섹션을 참조하세요. AWS CLI

### describe-maintenance-window-tasks

다음 코드 예시에서는 describe-maintenance-window-tasks을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 유지 관리 기간의 모든 작업을 나열하는 방법

다음 describe-maintenance-window-tasks 예제에서는 지정된 유지 관리 기간의 모든 작업을 나열합니다.

```
aws ssm describe-maintenance-window-tasks \
    --window-id "mw-06cf17cbefEXAMPLE"
```

출력:

```
{
    "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"
                    ]
                }
            ]
        }
    ]
}
```

예제 2: AWS-RunPowerShellScript command 문서를 호출하는 유지 관리 기간의 모든 작업을 나열하려면

다음 `describe-maintenance-window-tasks` 예제에서는 `AWS-RunPowerShellScript` 명령 문서를 간접 호출하는 지정된 유지 관리 기간의 모든 작업을 나열합니다.

```
aws ssm describe-maintenance-window-tasks \
    --window-id "mw-ab12cd34eEXAMPLE" \
    --filters "Key=TaskArn,Values=AWS-RunPowerShellScript"
```

출력:

```
{
    "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"
        }
    ]
}
```

예제 3: 우선순위가 3인 유지 관리 기간의 모든 작업을 나열하는 방법

다음 describe-maintenance-window-tasks 예제에서는 3이 Priority인 지정된 유지 관리 기간의 모든 작업을 나열합니다.

```
aws ssm describe-maintenance-window-tasks \
    --window-id "mw-ab12cd34eEXAMPLE" \
    --filters "Key=Priority,Values=3"
```

출력:

```
{
    "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"
        }
    ]
}
```

예제 4: 우선순위가 1이고 Run Command를 사용하는 유지 관리 기간의 모든 작업을 나열하는 방법

이 `describe-maintenance-window-tasks` 예제에서는 `1`이 `Priority`이고 `Run Command`를 사용하는 지정된 유지 관리 기간의 모든 작업을 나열합니다.

```
aws ssm describe-maintenance-window-tasks \
    --window-id "mw-ab12cd34eEXAMPLE" \
    --filters "Key=Priority,Values=1" "Key=TaskType,Values=RUN_COMMAND"
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 유지 관리 기간(AWS CLI)에 대한 정보 보기를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeMaintenanceWindowTasks의 섹션을 참조하세요. AWS CLI

### `describe-maintenance-windows-for-target`

다음 코드 예시에서는 `describe-maintenance-windows-for-target`을 사용하는 방법을 보여 줍니다.

AWS CLI

특정 인스턴스와 연결된 모든 유지 관리 기간을 나열하려면

다음 `describe-maintenance-windows-for-target` 예제에서는 지정된 인스턴스와 연결된 대상 또는 태스크가 있는 유지 관리 기간을 나열합니다.

```
aws ssm describe-maintenance-windows-for-target \
    --targets Key=InstanceIds,Values=i-1234567890EXAMPLE \
    --resource-type INSTANCE
```

출력:

```
{
    "WindowIdentities": [
        {
            "WindowId": "mw-0c5ed765acEXAMPLE",
            "Name": "My-First-Maintenance-Window"
        }
    ]
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 유지 관리 Windows(AWS CLI)에 대한 정보 보기를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeMaintenanceWindowsForTarget의 섹션을 참조하세요. AWS CLI

### `describe-maintenance-windows`

다음 코드 예시에서는 `describe-maintenance-windows`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 모든 유지 관리 기간을 나열하는 방법

다음 `describe-maintenance-windows` 예제에서는 현재 리전에 있는 AWS 계정의 모든 유지 관리 기간을 나열합니다.

```
aws ssm describe-maintenance-windows
```

출력:

```
{
    "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"
        },
    ]
}
```

예제 2: 활성화된 모든 유지 관리 기간을 나열하는 방법

다음 `describe-maintenance-windows` 예제에서는 활성화된 모든 유지 관리 기간을 나열합니다.

```
aws ssm describe-maintenance-windows \
    --filters "Key=Enabled,Values=true"
```

예제 3: 특정 이름과 일치하는 유지 관리 기간을 나열하는 방법

이 `describe-maintenance-windows` 예제에서는 지정된 이름의 모든 유지 관리 기간을 나열합니다.

```
aws ssm describe-maintenance-windows \
    --filters "Key=Name,Values=MyMaintenanceWindow"
```

자세한 내용은 AWS Systems Manager 사용 설명서의 유지 관리 Windows(AWS CLI)에 대한 정보 보기를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeMaintenanceWindows의 섹션을 참조하세요. AWS CLI

### `describe-ops-items`

다음 코드 예시에서는 `describe-ops-items`을 사용하는 방법을 보여 줍니다.

AWS CLI

의 세트를 나열하려면 OpsItems

다음 describe-ops-items 예제에서는 OpsItems AWS 계정에 열려 있는 모든 목록의 목록을 표시합니다.

```
aws ssm describe-ops-items \
    --ops-item-filters "Key=Status,Values=Open,Operator=Equal"
```

출력:

```
{
    "OpsItemSummaries": [
        {
            "CreatedBy": "arn:aws:sts::111222333444:assumed-role/OpsItem-CWE-Role/
fbf77cbe264a33509569f23e4EXAMPLE",
            "CreatedTime": "2020-03-14T17:02:46.375000-07:00",
            "LastModifiedBy": "arn:aws:sts::111222333444:assumed-role/OpsItem-CWE-
Role/fbf77cbe264a33509569f23e4EXAMPLE",
            "LastModifiedTime": "2020-03-14T17:02:46.375000-07:00",
            "Source": "SSM",
            "Status": "Open",
            "OpsItemId": "oi-7cfc5EXAMPLE",
            "Title": "SSM Maintenance Window execution failed",
            "OperationalData": {
                "/aws/dedup": {
                    "Value": "{\"dedupString\":\"SSMOpsItems-SSM-maintenance-window-
execution-failed\"}",
                    "Type": "SearchableString"
                },
                "/aws/resources": {
                    "Value": "[{\"arn\":\"arn:aws:ssm:us-
east-2:111222333444:maintenancewindow/mw-034093d322EXAMPLE\"}]",
                    "Type": "SearchableString"
                }
            },
            "Category": "Availability",
            "Severity": "3"
        },
        {
            "CreatedBy": "arn:aws:sts::1112223233444:assumed-role/OpsItem-CWE-Role/
fbf77cbe264a33509569f23e4EXAMPLE",
            "CreatedTime": "2020-02-26T11:43:15.426000-08:00",
            "LastModifiedBy": "arn:aws:sts::111222333444:assumed-role/OpsItem-CWE-
Role/fbf77cbe264a33509569f23e4EXAMPLE",
            "LastModifiedTime": "2020-02-26T11:43:15.426000-08:00",
```

```
            "Source": "EC2",
            "Status": "Open",
            "OpsItemId": "oi-6f966EXAMPLE",
            "Title": "EC2 instance stopped",
            "OperationalData": {
                "/aws/automations": {
                    "Value": "[ { \"automationType\": \"AWS:SSM:Automation\",
 \"automationId\": \"AWS-RestartEC2Instance\" } ]",
                    "Type": "SearchableString"
                },
                "/aws/dedup": {
                    "Value": "{\"dedupString\":\"SSMOpsItems-EC2-instance-stopped
\"}",
                    "Type": "SearchableString"
                },
                "/aws/resources": {
                    "Value": "[{\"arn\":\"arn:aws:ec2:us-
east-2:111222333444:instance/i-0beccfbc02EXAMPLE\"}]",
                    "Type": "SearchableString"
                }
            },
            "Category": "Availability",
            "Severity": "3"
        }
    ]
}
```

자세한 내용은 AWS Systems Manager 사용 설명서 의 작업 OpsItems 단원을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeOpsItems의 섹션을 참조하세요. AWS CLI

### describe-parameters

다음 코드 예시에서는 describe-parameters을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: 모든 파라미터를 나열하는 방법

다음 describe-parameters 예제에서는 현재 AWS 계정 및 리전의 모든 파라미터를 나열합니다.

```
aws ssm describe-parameters
```

출력:

```
{
    "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": []
        }
    ]
}
```

예 2: 특정 메타데이터와 일치하는 모든 파라미터를 나열하는 방법

이 `describe-parameters` 예시에서는 필터와 일치하는 모든 파라미터를 나열합니다.

aws ssm describe-parameters --filters 'Key=Type,Values=StringList'

출력:

```
{
    "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": []
        }
    ]
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 Systems Manager 파라미터 검색을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeParameters의 섹션을 참조하세요. AWS CLI

### `describe-patch-baselines`

다음 코드 예시에서는 `describe-patch-baselines`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 모든 패치 기준을 나열하는 방법

다음 `describe-patch-baselines` 예제에서는 현재 리전의 계정에서 모든 패치 기준에 대한 세부 정보를 가져옵니다.

```
aws ssm describe-patch-baselines
```

출력:

```
{
    "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"
        }
    ]
}
```

예제 2: 에서 제공하는 모든 패치 기준을 나열하려면 AWS

다음 `describe-patch-baselines` 예제에서는 에서 제공하는 모든 패치 기준을 나열합니다 AWS.

```
aws ssm describe-patch-baselines \
    --filters "Key=OWNER,Values=[AWS]"
```

예제 3: 소유한 모든 패치 기준을 나열하는 방법

다음 `describe-patch-baselines` 예제에서는 현재 리전의 계정에서 생성된 모든 사용자 지정 패치 기준을 나열합니다.

```
aws ssm describe-patch-baselines \
    --filters "Key=OWNER,Values=[Self]"
```

자세한 내용은 AWS Systems Manager 사용 설명서의 사전 정의된 패치 기준 및 사용자 지정 패치 기준 정보를 참조하세요.

- 자세한 API 내용은 명령 참조DescribePatchBaselines의 섹션을 참조하세요. AWS CLI

### describe-patch-group-state

다음 코드 예시에서는 `describe-patch-group-state`을 사용하는 방법을 보여 줍니다.

AWS CLI

패치 그룹의 상태를 가져오는 방법

다음 `describe-patch-group-state` 예제에서는 패치 그룹에 대한 개요 수준의 패치 규정 준수 요약을 검색합니다.

```
aws ssm describe-patch-group-state \
    --patch-group "Production"
```

출력:

```
{
    "Instances": 21,
    "InstancesWithCriticalNonCompliantPatches": 1,
    "InstancesWithFailedPatches": 2,
    "InstancesWithInstalledOtherPatches": 3,
    "InstancesWithInstalledPatches": 21,
    "InstancesWithInstalledPendingRebootPatches": 2,
    "InstancesWithInstalledRejectedPatches": 1,
    "InstancesWithMissingPatches": 3,
    "InstancesWithNotApplicablePatches": 4,
    "InstancesWithOtherNonCompliantPatches": 1,
    "InstancesWithSecurityNonCompliantPatches": 1,
    "InstancesWithUnreportedNotApplicablePatches": 2
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 패치 그룹 정보 <https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-patchgroups.html>__ 및 패치 규정 준수 상태 값 이해를 참조하세요.

- 자세한 API 내용은 명령 참조DescribePatchGroupState의 섹션을 참조하세요. AWS CLI

### describe-patch-groups

다음 코드 예시에서는 describe-patch-groups을 사용하는 방법을 보여 줍니다.

AWS CLI

패치 그룹 등록을 표시하는 방법

다음 describe-patch-groups 예제에서는 패치 그룹 등록을 나열합니다.

```
aws ssm describe-patch-groups
```

출력:

```
{
    "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
            }
        },
        ...
    ]
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 패치 그룹 생성 <https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-group-tagging.html>__ 및 패치 기준에 패치 그룹 추가를 참조하세요.

- 자세한 API 내용은 명령 참조DescribePatchGroups의 섹션을 참조하세요. AWS CLI

### describe-patch-properties

다음 코드 예시에서는 describe-patch-properties을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon Linux 패치 가용성을 나열하려면

다음 describe-patch-properties 예제에서는 AWS 계정에서 패치를 사용할 수 있는 Amazon Linux 제품의 목록을 보여줍니다.

```
aws ssm describe-patch-properties \
    --operating-system AMAZON_LINUX \
    --property PRODUCT
```

출력:


```
{
    "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"
        }
    ]
```

```
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 패치 기준 정보를 참조하세요.

- 자세한 API 내용은 명령 참조DescribePatchProperties의 섹션을 참조하세요. AWS CLI

### describe-sessions

다음 코드 예시에서는 describe-sessions을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 모든 활성 Session Manager 세션을 나열하려면

이 describe-sessions 예제에서는 지정된 사용자가 시작한 지난 30일 동안 가장 최근에 생성된 활성 세션(연결된 세션과 연결 해제된 세션 모두)의 목록을 검색합니다. 이 명령은 Session Manager를 사용하여 시작된 대상에 대한 연결 결과만 반환합니다. 원격 데스크톱 연결 또는 와 같은 다른 수단을 통한 연결은 나열되지 않습니다SSH.

```
aws ssm describe-sessions \
    --state "Active" \
    --filters "key=Owner,value=arn:aws:sts::123456789012:assumed-role/Administrator/
Shirley-Rodriguez"
```

출력:

```
{
    "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": {}
        }
    ]
}
```

예제 2: 종료된 모든 Session Manager 세션을 나열하려면

이 `describe-sessions` 예제에서는 모든 사용자에 대해 지난 30일 동안 가장 최근에 종료된 세션 목록을 검색합니다.

```
aws ssm describe-sessions \
    --state "History"
```

출력:

```
{
    "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--"
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 세션 기록 보기를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeSessions의 섹션을 참조하세요. AWS CLI

### `disassociate-ops-item-related-item`

다음 코드 예시에서는 `disassociate-ops-item-related-item`을 사용하는 방법을 보여 줍니다.

AWS CLI

관련 항목 연결을 삭제하려면

다음 `disassociate-ops-item-related-item` 예제에서는 OpsItem 와 관련 항목 간의 연결을 삭제합니다.

```
aws ssm disassociate-ops-item-related-item \
    --ops-item-id "oi-f99f2EXAMPLE" \
    --association-id "e2036148-cccb-490e-ac2a-390e5EXAMPLE"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Systems Manager 사용 설명서의 에서 Incident Manager 인시던트 작업을 OpsCenter 참조하세요.

- 자세한 API 내용은 명령 참조DisassociateOpsItemRelatedItem의 섹션을 참조하세요. AWS CLI

### `get-automation-execution`

다음 코드 예시에서는 `get-automation-execution`을 사용하는 방법을 보여 줍니다.

AWS CLI

자동화 실행에 대한 세부 정보를 표시하는 방법

다음 `get-automation-execution` 예제에서는 자동화 실행에 대한 세부 정보를 표시합니다.

```
aws ssm get-automation-execution \
    --automation-execution-id 73c8eef8-f4ee-4a05-820c-e354fEXAMPLE
```

출력:

```
{
    "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
        }
    }
```

```
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 연습: Linux 패치AMI(AWS CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조GetAutomationExecution의 섹션을 참조하세요. AWS CLI

### get-calendar-state

다음 코드 예시에서는 get-calendar-state을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 변경 달력의 현재 상태를 가져오려면

이 get-calendar-state 예제는 현재 시점의 달력 상태를 반환합니다. 예제에서는 시간을 지정하지 않으므로 달력의 현재 상태가 보고됩니다.

```
aws ssm get-calendar-state \
    --calendar-names "MyCalendar"
```

출력:

```
{
    "State": "OPEN",
    "AtTime": "2020-02-19T22:28:51Z",
    "NextTransitionTime": "2020-02-24T21:15:19Z"
}
```

예제 2: 지정된 시간에 변경 달력의 상태를 가져오려면

이 get-calendar-state 예제는 지정된 시간에 일정 상태를 반환합니다.

```
aws ssm get-calendar-state \
    --calendar-names "MyCalendar" \
    --at-time "2020-07-19T21:15:19Z"
```

출력:

```
{
    "State": "CLOSED",
```

```
    "AtTime": "2020-07-19T21:15:19Z"
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 변경 달력 상태 가져오기를 참조하세요.

- 자세한 API 내용은 명령 참조GetCalendarState의 섹션을 참조하세요. AWS CLI

### get-command-invocation

다음 코드 예시에서는 get-command-invocation을 사용하는 방법을 보여 줍니다.

AWS CLI

명령 간접 호출의 세부 정보를 표시하는 방법

다음 get-command-invocation 예제에서는 지정된 인스턴스에서 지정된 명령의 모든 간접 호출을 나열합니다.

```
aws ssm get-command-invocation \
    --command-id "ef7fdfd8-9b57-4151-a15c-db9a12345678" \
    --instance-id "i-1234567890abcdef0"
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 명령 상태 이해를 참조하세요.

- 자세한 API 내용은 명령 참조GetCommandInvocation의 섹션을 참조하세요. AWS CLI

### get-connection-status

다음 코드 예시에서는 get-connection-status을 사용하는 방법을 보여 줍니다.

AWS CLI

관리형 인스턴스의 연결 상태를 표시하는 방법

이 get-connection-status 예제에서는 지정된 관리형 인스턴스의 연결 상태를 반환합니다.

```
aws ssm get-connection-status \
    --target i-1234567890abcdef0
```

출력:

```
{
    "Target": "i-1234567890abcdef0",
    "Status": "connected"
}
```

- 자세한 API 내용은 명령 참조GetConnectionStatus의 섹션을 참조하세요. AWS CLI

### get-default-patch-baseline

다음 코드 예시에서는 get-default-patch-baseline을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 기본 Windows 패치 기준을 표시하는 방법

다음 get-default-patch-baseline 예제에서는 Windows Server의 기본 패치 기준에 대한 세부 정보를 검색합니다.

```
aws ssm get-default-patch-baseline
```

출력:

```
{
  "BaselineId": "pb-0713accee01612345",
  "OperatingSystem": "WINDOWS"
}
```

예제 2: Amazon Linux의 기본 패치 기준을 표시하는 방법

다음 get-default-patch-baseline 예제에서는 Amazon Linux의 기본 패치 기준에 대한 세부 정보를 검색합니다.

```
aws ssm get-default-patch-baseline \
    --operating-system AMAZON_LINUX
```

출력:

```
{
    "BaselineId": "pb-047c6eb9c8fc12345",
    "OperatingSystem": "AMAZON_LINUX"
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 사전 정의 및 사용자 지정 패치 기준 <https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-baselines.html>__ 정보 및 기존 패치 기준을 기본값으로 설정을 참조하세요.

- 자세한 API 내용은 명령 참조GetDefaultPatchBaseline의 섹션을 참조하세요. AWS CLI

### get-deployable-patch-snapshot-for-instance

다음 코드 예시에서는 get-deployable-patch-snapshot-for-instance을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스에서 사용하는 패치 기준에 대한 현재 스냅샷을 검색하는 방법

다음 get-deployable-patch-snapshot-for-instance 예제에서는 인스턴스에서 사용하는 지정된 패치 기준의 현재 스냅샷에 대한 세부 정보를 검색합니다. 이 명령은 인스턴스 자격 증명을 사용하여 인스턴스에서 실행해야 합니다. 인스턴스 자격 증명을 사용하도록 하려면 aws configure를 실행하고 인스턴스의 리전만 지정합니다. Access Key 및 Secret Key 필드는 비워 둡니다.

팁: uuidgen을 사용하여 snapshot-id를 생성합니다.

```
aws ssm get-deployable-patch-snapshot-for-instance \
    --instance-id "i-1234567890abcdef0" \
    --snapshot-id "521c3536-930c-4aa9-950e-01234567abcd"
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 파라미터 이름: 스냅샷 ID를 참조하세요.

- 자세한 API 내용은 명령 참조GetDeployablePatchSnapshotForInstance의 섹션을 참조하세요. AWS CLI

### get-document

다음 코드 예시에서는 get-document을 사용하는 방법을 보여 줍니다.

AWS CLI

문서 콘텐츠를 가져오는 방법

다음 get-document 예제에서는 Systems Manager 문서의 콘텐츠를 표시합니다.

```
aws ssm get-document \
    --name "AWS-RunShellScript"
```

출력:

```
{
    "Name": "AWS-RunShellScript",
    "DocumentVersion": "1",
    "Status": "Active",
    "Content": "{\n    \"schemaVersion\":\"1.2\",\n    \"description\":\"Run a
 shell script or specify the commands to run.\",\n    \"parameters\":{\n
 \"commands\":{\n            \"type\":\"StringList\",\n            \"description
\":\"(Required) Specify a shell script or a command to run.\",\n
 \"minItems\":1,\n            \"displayType\":\"textarea\"\n        },\n
 \"workingDirectory\":{\n            \"type\":\"String\",\n            \"default
\":\"\",\n            \"description\":\"(Optional) The path to the working
 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
 command to complete before it is considered to have failed. Default is 3600 (1
 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",
    "DocumentFormat": "JSON"
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 AWS Systems Manager 문서를 참조하세요.

- 자세한 API 내용은 명령 참조GetDocument의 섹션을 참조하세요. AWS CLI

### `get-inventory-schema`

다음 코드 예시에서는 `get-inventory-schema`을 사용하는 방법을 보여 줍니다.

AWS CLI

인벤토리 스키마를 보는 방법

이 예제에서는 계정의 인벤토리 유형 이름 목록을 반환합니다.

명령:

```
aws ssm get-inventory-schema
```

출력:

```
{
  "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--"
}
```

특정 인벤토리 유형의 인벤토리 스키마를 보는 방법

이 예제에서는 AWS:AWS Component 인벤토리 유형에 대한 인벤토리 스키마를 반환합니다.

명령:

```
aws ssm get-inventory-schema --type-name "AWS:AWSComponent"
```

- 자세한 API 내용은 명령 참조GetInventorySchema의 섹션을 참조하세요. AWS CLI

### `get-inventory`

다음 코드 예시에서는 `get-inventory`을 사용하는 방법을 보여 줍니다.

AWS CLI

인벤토리 페이지를 보는 방법

이 예제에서는 인벤토리의 사용자 지정 메타데이터를 가져옵니다.

명령:

```
aws ssm get-inventory
```

출력:

```
{
    "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"
    }
  ]
}
```

- 자세한 API 내용은 명령 참조GetInventory의 섹션을 참조하세요. AWS CLI

### get-maintenance-window-execution-task-invocation

다음 코드 예시에서는 get-maintenance-window-execution-task-invocation을 사용하는 방법을 보여 줍니다.

AWS CLI

유지 관리 기간 작업 호출에 대한 정보를 가져오려면

다음 get-maintenance-window-execution-task-invocation 예제에서는 지정된 유지 관리 기간 실행의 일부인 지정된 작업 호출에 대한 정보를 나열합니다.

```
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"
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 태스크 및 태스크 실행에 대한 정보 보기 (AWS CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조GetMaintenanceWindowExecutionTaskInvocation의 섹션을 참조하세요. AWS CLI

### get-maintenance-window-execution-task

다음 코드 예시에서는 get-maintenance-window-execution-task을 사용하는 방법을 보여 줍니다.

AWS CLI

유지 관리 기간 작업 실행에 대한 정보를 가져오는 방법

다음 get-maintenance-window-execution-task 예제에서는 지정된 유지 관리 기간 실행의 일부인 작업에 대한 정보를 나열합니다.

```
aws ssm get-maintenance-window-execution-task \
    --window-execution-id "518d5565-5969-4cca-8f0e-da3b2EXAMPLE" \
    --task-id "ac0c6ae1-daa3-4a89-832e-d3845EXAMPLE"
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 태스크 및 태스크 실행(AWS CLI)에 대한 정보 보기를 참조하세요.

- 자세한 API 내용은 명령 참조GetMaintenanceWindowExecutionTask의 섹션을 참조하세요. AWS CLI

### `get-maintenance-window-execution`

다음 코드 예시에서는 `get-maintenance-window-execution`을 사용하는 방법을 보여 줍니다.

AWS CLI

유지 관리 기간 작업 실행에 대한 정보를 가져오는 방법

다음 `get-maintenance-window-execution` 예제에서는 지정된 유지 관리 기간 실행의 일부로 실행된 작업에 대한 정보를 나열합니다.

```
aws ssm get-maintenance-window-execution \
    --window-execution-id "518d5565-5969-4cca-8f0e-da3b2EXAMPLE"
```

출력:

```
{
    "Status": "SUCCESS",
    "TaskIds": [
        "ac0c6ae1-daa3-4a89-832e-d3845EXAMPLE"
    ],
    "StartTime": 1487692834.595,
    "EndTime": 1487692835.051,
    "WindowExecutionId": "518d5565-5969-4cca-8f0e-da3b2EXAMPLE",
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 태스크 및 태스크 실행에 대한 정보 보기(AWS CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조GetMaintenanceWindowExecution의 섹션을 참조하세요. AWS CLI

### get-maintenance-window-task

다음 코드 예시에서는 get-maintenance-window-task을 사용하는 방법을 보여 줍니다.

AWS CLI

유지 관리 기간 작업에 대한 정보를 가져오려면

다음 get-maintenance-window-task 예제에서는 지정된 유지 관리 기간 작업에 대한 세부 정보를 검색합니다.

```
aws ssm get-maintenance-window-task \
    --window-id mw-0c5ed765acEXAMPLE \
    --window-task-id 0e842a8d-2d44-4886-bb62-af8dcEXAMPLE
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 유지 관리 Windows(AWS CLI)에 대한 정보 보기를 참조하세요.

- 자세한 API 내용은 명령 참조GetMaintenanceWindowTask의 섹션을 참조하세요. AWS CLI

### get-maintenance-window

다음 코드 예시에서는 get-maintenance-window을 사용하는 방법을 보여 줍니다.

AWS CLI

유지 관리 기간에 대한 정보를 가져오는 방법

다음 get-maintenance-window 예제에서는 지정된 유지 관리 기간에 대한 세부 정보를 검색합니다.

```
aws ssm get-maintenance-window \
    --window-id "mw-03eb9db428EXAMPLE"
```

출력:

```
{
    "AllowUnassociatedTargets": true,
    "CreatedDate": 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"
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 유지 관리 기간(AWS CLI)에 대한 정보 보기를 참조하세요.

- 자세한 API 내용은 명령 참조GetMaintenanceWindow의 섹션을 참조하세요. AWS CLI

### get-ops-item

다음 코드 예시에서는 get-ops-item을 사용하는 방법을 보여 줍니다.

AWS CLI

에 대한 정보를 보려면 OpsItem

다음 get-ops-item 예제에서는 지정된 에 대한 세부 정보를 표시합니다 OpsItem.

```
aws ssm get-ops-item \
    --ops-item-id oi-0b725EXAMPLE
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS Systems Manager 사용 설명서 의 작업을 OpsItems 참조하세요.

- 자세한 API 내용은 명령 참조GetOpsItem의 섹션을 참조하세요. AWS CLI

### `get-ops-summary`

다음 코드 예시에서는 `get-ops-summary`을 사용하는 방법을 보여 줍니다.

AWS CLI

전체 요약을 보려면 OpsItems

다음 `get-ops-summary` 예제에서는 AWS 계정의 모든 OpsItems 에 대한 요약을 표시합니다.

```
aws ssm get-ops-summary
```

출력:

```
{
    "Entities": [
        {
            "Id": "oi-4309fEXAMPLE",
            "Data": {
                "AWS:OpsItem": {
                    "CaptureTime": "2020-02-26T18:58:32.918Z",
                    "Content": [
                        {
                            "AccountId": "111222333444",
                            "Category": "Availability",
                            "CreatedBy": "arn:aws:sts::111222333444:assumed-role/
OpsItem-CWE-Role/fbf77cbe264a33509569f23e4EXAMPLE",
                            "CreatedTime": "2020-02-26T19:10:44.149Z",
                            "Description": "CloudWatch Event Rule SSMOpsItems-EC2-
instance-terminated was triggered. Your EC2 instance has terminated. See below for
 more details.",
                            "LastModifiedBy": "arn:aws:sts::111222333444:assumed-
role/OpsItem-CWE-Role/fbf77cbe264a33509569f23e4EXAMPLE",
                            "LastModifiedTime": "2020-02-26T19:10:44.149Z",
                            "Notifications": "",
                            "OperationalData": "{\"/aws/automations\":
{\"type\":\"SearchableString\",\"value\":\"[ { \\\"automationType\\\": \\
\"AWS:SSM:Automation\\\", \\\"automationId\\\": \\\"AWS-CreateManagedWindowsInstance
\\\" }, { \\\"automationType\\\": \\\"AWS:SSM:Automation\\\", \\\"automationId
\\\": \\\"AWS-CreateManagedLinuxInstance\\\" } ]\"},\"/aws/resources\":
{\"type\":\"SearchableString\",\"value\":\"[{\\\"arn\\\":\\\"arn:aws:ec2:us-
east-2:111222333444:instance/i-0acbd0800fEXAMPLE\\\"}]\"},\"/aws/dedup\":{\"type\":
\"SearchableString\",\"value\":\"{\\\"dedupString\\\":\\\"SSMOpsItems-EC2-instance-
terminated\\\"}\"}}",
                            "OpsItemId": "oi-4309fEXAMPLE",
                            "RelatedItems": "",
                            "Severity": "3",
                            "Source": "EC2",
                            "Status": "Open",
                            "Title": "EC2 instance terminated"
                        }
                    ]
                }
```

```
            }
        },
        {
            "Id": "oi-bb2a0e6a4541",
            "Data": {
                "AWS:OpsItem": {
                    "CaptureTime": "2019-11-26T19:20:06.161Z",
                    "Content": [
                        {
                            "AccountId": "111222333444",
                            "Category": "Availability",
                            "CreatedBy": "arn:aws:sts::111222333444:assumed-role/
OpsItem-CWE-Role/fbf77cbe264a33509569f23e4EXAMPLE",
                            "CreatedTime": "2019-11-26T20:00:07.237Z",
                            "Description": "CloudWatch Event Rule SSMOpsItems-SSM-
maintenance-window-execution-failed was triggered. Your SSM Maintenance Window
 execution has failed. See below for more details.",
                            "LastModifiedBy": "arn:aws:sts::111222333444:assumed-
role/OpsItem-CWE-Role/fbf77cbe264a33509569f23e4EXAMPLE",
                            "LastModifiedTime": "2019-11-26T20:00:07.237Z",
                            "Notifications": "",
                            "OperationalData": "{\"/aws/resources\":{\"type
\":\"SearchableString\",\"value\":\"[{\\\"arn\\\":\\\"arn:aws:ssm:us-
east-2:111222333444:maintenancewindow/mw-0e83ba440dEXAMPLE\\\"}]\"},\"/aws/dedup\":
{\"type\":\"SearchableString\",\"value\":\"{\\\"dedupString\\\":\\\"SSMOpsItems-SSM-
maintenance-window-execution-failed\\\"}\"}}",
                            "OpsItemId": "oi-bb2a0EXAMPLE",
                            "RelatedItems": "",
                            "Severity": "3",
                            "Source": "SSM",
                            "Status": "Open",
                            "Title": "SSM Maintenance Window execution failed"
                        }
                    ]
                }
            }
        }
    ]
}
```

자세한 내용은 AWS Systems Manager 사용 설명서 의 작업을 OpsItems 참조하세요.

- 자세한 API 내용은 명령 참조GetOpsSummary의 섹션을 참조하세요. AWS CLI

### get-parameter-history

다음 코드 예시에서는 get-parameter-history을 사용하는 방법을 보여 줍니다.

AWS CLI

파라미터 값 기록을 가져오는 방법

다음 get-parameter-history 예제에서는 해당 값을 포함하여 지정된 파라미터의 변경 기록을 나열합니다.

```
aws ssm get-parameter-history \
    --name "MyStringParameter"
```

출력:

```
{
    "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": []
        }
    ]
}
```

자세한 내용은 AWS  Systems Manager 사용 설명서의 파라미터 버전 작업을 참조하세요.

- 자세한 API 내용은 명령 참조GetParameterHistory의 섹션을 참조하세요. AWS CLI

### `get-parameter`

다음 코드 예시에서는 `get-parameter`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 파라미터 값을 표시하는 방법

다음 `get-parameter` 예제에서는 지정된 단일 파라미터의 값을 나열합니다.

```
aws ssm get-parameter \
    --name "MyStringParameter"
```

출력:

```
{
    "Parameter": {
        "Name": "MyStringParameter",
        "Type": "String",
        "Value": "Veni",
        "Version": 1,
        "LastModifiedDate": 1530018761.888,
        "ARN": "arn:aws:ssm:us-east-2:111222333444:parameter/MyStringParameter"
        "DataType": "text"
    }
```

```
}
```

자세한 내용은 AWS  Systems Manager 사용 설명서의 Parameter Store 작업을 참조하세요.

예제 2: SecureString 파라미터 값을 복호화하려면

다음 `get-parameter` 예제에서는 지정된 `SecureString` 파라미터의 값을 해독합니다.

```
aws ssm get-parameter \
    --name "MySecureStringParameter" \
    --with-decryption
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS  Systems Manager 사용 설명서의 Parameter Store 작업을 참조하세요.

예제 3: 레이블을 사용하여 파라미터 값을 표시하는 방법

다음 `get-parameter` 예제에서는 지정된 레이블을 포함하는 지정된 단일 파라미터 값을 나열합니다.

```
aws ssm get-parameter \
    --name "MyParameter:label"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 파라미터 레이블 작업을 참조하세요.

예제 4: 버전을 사용하여 파라미터 값을 표시하는 방법

다음 `get-parameter` 예제에서는 지정된 단일 파라미터 버전의 값을 나열합니다.

```
aws ssm get-parameter \
    --name "MyParameter:2"
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 파라미터 레이블 작업을 참조하세요.

- 자세한 API 내용은 명령 참조GetParameter의 섹션을 참조하세요. AWS CLI

### get-parameters-by-path

다음 코드 예시에서는 `get-parameters-by-path`을 사용하는 방법을 보여 줍니다.

AWS CLI

특정 경로의 파라미터를 나열하려면

다음 get-parameters-by-path 예제에서는 지정된 계층 구조 내의 파라미터를 나열합니다.

```
aws ssm get-parameters-by-path \
    --path "/site/newyork/department/"
```

출력:

```
{
    "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"
        },
        ...
    ]
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 파라미터 계층 작업을 참조하세요.

- 자세한 API 내용은 명령 참조GetParametersByPath의 섹션을 참조하세요. AWS CLI

### get-parameters

다음 코드 예시에서는 get-parameters을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 파라미터 값을 나열하는 방법

다음 get-parameters 예제에서는 지정된 세 개의 파라미터 값을 나열합니다.

```
aws ssm get-parameters \
    --names "MyStringParameter" "MyStringListParameter" "MyInvalidParameterName"
```

출력:

```
{
    "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"
    ]
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 Parameter Store 작업을 참조하세요.

예제 2: "--query" 옵션을 사용하여 여러 파라미터의 이름과 값을 나열하는 방법

다음 get-parameters 예제에서는 지정된 파라미터의 이름 및 값을 나열합니다.

```
aws ssm get-parameters \
    --names MyStringParameter MyStringListParameter \
    --query "Parameters[*].{Name:Name,Value:Value}"
```

출력:

```
[
    {
        "Name": "MyStringListParameter",
        "Value": "alpha,beta,gamma"
    },
    {
        "Name": "MyStringParameter",
        "Value": "Vidi"
    }
]
```

자세한 내용은 AWS Systems Manager 사용 설명서의 Parameter Store 작업을 참조하세요.

예제 3: 레이블을 사용하여 파라미터 값을 표시하는 방법

다음 get-parameter 예제에서는 지정된 레이블을 포함하는 지정된 단일 파라미터 값을 나열합니다.

```
aws ssm get-parameter \
    --name "MyParameter:label"
```

출력:

```
{
    "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": []
}
```

자세한 내용은 AWS  Systems Manager 사용 설명서의 파라미터 레이블 작업을 참조하세요.

- 자세한 API 내용은 명령 참조GetParameters의 섹션을 참조하세요. AWS CLI

### get-patch-baseline-for-patch-group

다음 코드 예시에서는 get-patch-baseline-for-patch-group을 사용하는 방법을 보여 줍니다.

AWS CLI

패치 그룹의 패치 기준을 표시하는 방법

다음 get-patch-baseline-for-patch-group 예제에서는 지정된 패치 그룹의 패치 기준에 대한 세부 정보를 검색합니다.

```
aws ssm get-patch-baseline-for-patch-group \
    --patch-group "DEV"
```

출력:

```
{
    "PatchGroup": "DEV",
    "BaselineId": "pb-0123456789abcdef0",
    "OperatingSystem": "WINDOWS"
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 패치 그룹 생성 <https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-group-tagging.html>__ 및 패치 기준에 패치 그룹 추가를 참조하세요.

- 자세한 API 내용은 명령 참조GetPatchBaselineForPatchGroup의 섹션을 참조하세요. AWS CLI

### get-patch-baseline

다음 코드 예시에서는 get-patch-baseline을 사용하는 방법을 보여 줍니다.

AWS CLI

패치 기준을 표시하는 방법

다음 get-patch-baseline 예제에서는 지정된 패치 기준에 대한 세부 정보를 검색합니다.

```
aws ssm get-patch-baseline \
    --baseline-id "pb-0123456789abcdef0"
```

출력:

```
{
    "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"
    ],
    "CreatedDate": 1550244180.465,
    "ModifiedDate": 1550244180.465,
    "Description": "Patches for Windows Servers",
    "Sources": []
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 패치 기준 정보를 참조하세요.

- 자세한 API 내용은 명령 참조GetPatchBaseline의 섹션을 참조하세요. AWS CLI

### get-service-setting

다음 코드 예시에서는 get-service-setting을 사용하는 방법을 보여 줍니다.

AWS CLI

Parameter Store 처리량에 대한 서비스 설정을 검색하려면

다음 get-service-setting 예제에서는 지정된 리전의 Parameter Store 처리량에 대한 현재 서비스 설정을 검색합니다.

```
aws ssm get-service-setting \
    --setting-id arn:aws:ssm:us-east-1:123456789012:servicesetting/ssm/parameter-store/high-throughput-enabled
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 파라미터 스토어 처리량 증가를 참조하세요.

- 자세한 API 내용은 명령 참조GetServiceSetting의 섹션을 참조하세요. AWS CLI

### `label-parameter-version`

다음 코드 예시에서는 `label-parameter-version`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 파라미터의 최신 버전에 레이블을 추가하려면

다음 `label-parameter-version` 예제에서는 지정된 파라미터의 최신 버전에 레이블을 추가합니다.

```
aws ssm label-parameter-version \
    --name "MyStringParameter" \
    --labels "ProductionReady"
```

출력:

```
{
    "InvalidLabels": [],
    "ParameterVersion": 3
}
```

자세한 내용은 AWS  Systems Manager 사용 설명서의 파라미터 레이블 작업을 참조하세요.

예제 2: 파라미터의 특정 버전에 레이블을 추가하려면

다음 `label-parameter-version` 예제에서는 지정된 버전의 파라미터에 레이블을 추가합니다.

```
aws ssm label-parameter-version \
    --name "MyStringParameter" \
    --labels "ProductionReady" \
    --parameter-version "2" --labels "DevelopmentReady"
```

자세한 내용은 AWS Systems Manager 사용 설명서의 파라미터 레이블 작업을 참조하세요.

- 자세한 API 내용은 명령 참조LabelParameterVersion의 섹션을 참조하세요. AWS CLI

### list-association-versions

다음 코드 예시에서는 list-association-versions을 사용하는 방법을 보여 줍니다.

AWS CLI

특정 연결 ID의 모든 연결 버전을 가져오는 방법

다음 list-association-versions 예제에서는 지정된 연결의 모든 버전을 나열합니다.

```
aws ssm list-association-versions \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

출력:

```
{
"AssociationVersions": [
        {
            "AssociationId": "8dfe3659-4309-493a-8755-0123456789ab",
            "AssociationVersion": "1",
            "CreatedDate": 1550505536.726,
            "Name": "AWS-UpdateSSMAgent",
            "Parameters": {
                "allowDowngrade": [
                    "false"
                ],
                "version": [
                    ""
                ]
            },
            "Targets": [
                {
                    "Key": "InstanceIds",
                    "Values": [
                        "i-1234567890abcdef0"
                    ]
                }
            ],
            "ScheduleExpression": "cron(0 00 12 ? * SUN *)",
```

```
            "AssociationName": "UpdateSSMAgent"
        }
    ]
}
```

자세한 내용은 AWS  Systems Manager 사용 설명서의 Systems Manager에서 연결 작업을 참조하세요.

- 자세한 API 내용은 명령 참조ListAssociationVersions의 섹션을 참조하세요. AWS CLI

### `list-associations`

다음 코드 예시에서는 `list-associations`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 특정 인스턴스의 연결을 나열하는 방법

다음 list-associations 예제에서는 AssociationName, U 와의 모든 연결을 나열합니다 pdateSSMAgent.

```
aws ssm list-associations /
    --association-filter-list "key=AssociationName,value=UpdateSSMAgent"
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은  Systems Manager 사용 설명서의 Systems Manager에서 연결 작업을 참조하세요.

예제 2: 특정 문서의 연결을 나열하는 방법

다음 list-associations 예제에서는 지정된 문서의 모든 연결을 나열합니다.

```
aws ssm list-associations /
    --association-filter-list "key=Name,value=AWS-UpdateSSMAgent"
```

출력:

```
{
    "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
                }
            }
        }
    ]
}
```

자세한 내용은  Systems Manager 사용 설명서의 Systems Manager에서 연결 작업을 참조하세요.

- 자세한 API 내용은 명령 참조ListAssociations의 섹션을 참조하세요. AWS CLI

### `list-command-invocations`

다음 코드 예시에서는 `list-command-invocations`을 사용하는 방법을 보여 줍니다.

AWS CLI

특정 명령의 간접 호출을 나열하는 방법

다음 `list-command-invocations` 예제에서는 명령의 모든 간접 호출을 나열합니다.

```
aws ssm list-command-invocations \
    --command-id "ef7fdfd8-9b57-4151-a15c-db9a12345678" \
    --details
```

출력:

```
{
    "CommandInvocations": [
        {
            "CommandId": "ef7fdfd8-9b57-4151-a15c-db9a12345678",
            "InstanceId": "i-02573cafcfEXAMPLE",
            "InstanceName": "",
            "Comment": "b48291dd-ba76-43e0-
b9df-13e11ddaac26:6960febb-2907-4b59-8e1a-d6ce8EXAMPLE",
            "DocumentName": "AWS-UpdateSSMAgent",
            "DocumentVersion": "",
            "RequestedDateTime": 1582136283.089,
            "Status": "Success",
            "StatusDetails": "Success",
            "StandardOutputUrl": "",
            "StandardErrorUrl": "",
            "CommandPlugins": [
                {
                    "Name": "aws:updateSsmAgent",
                    "Status": "Success",
                    "StatusDetails": "Success",
                    "ResponseCode": 0,
                    "ResponseStartDateTime": 1582136283.419,
                    "ResponseFinishDateTime": 1582136283.51,
                    "Output": "Updating amazon-ssm-agent from 2.3.842.0 to latest
\nSuccessfully downloaded https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/
ssm-agent-manifest.json\namazon-ssm-agent 2.3.842.0 has already been installed,
 update skipped\n",
                    "StandardOutputUrl": "",
                    "StandardErrorUrl": "",
                    "OutputS3Region": "us-east-2",
                    "OutputS3BucketName": "",
                    "OutputS3KeyPrefix": ""
                }
            ],
            "ServiceRole": "",
            "NotificationConfig": {
                "NotificationArn": "",
```

```
                "NotificationEvents": [],
                "NotificationType": ""
            },
            "CloudWatchOutputConfig": {
                "CloudWatchLogGroupName": "",
                "CloudWatchOutputEnabled": false
            }
        },
        {
            "CommandId": "ef7fdfd8-9b57-4151-a15c-db9a12345678",
            "InstanceId": "i-0471e04240EXAMPLE",
            "InstanceName": "",
            "Comment": "b48291dd-ba76-43e0-
b9df-13e11ddaac26:6960febb-2907-4b59-8e1a-d6ce8EXAMPLE",
            "DocumentName": "AWS-UpdateSSMAgent",
            "DocumentVersion": "",
            "RequestedDateTime": 1582136283.02,
            "Status": "Success",
            "StatusDetails": "Success",
            "StandardOutputUrl": "",
            "StandardErrorUrl": "",
            "CommandPlugins": [
                {
                    "Name": "aws:updateSsmAgent",
                    "Status": "Success",
                    "StatusDetails": "Success",
                    "ResponseCode": 0,
                    "ResponseStartDateTime": 1582136283.812,
                    "ResponseFinishDateTime": 1582136295.031,
                    "Output": "Updating amazon-ssm-agent from 2.3.672.0 to latest
\nSuccessfully downloaded https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/
ssm-agent-manifest.json\nSuccessfully downloaded https://s3.us-east-2.amazonaws.com/
amazon-ssm-us-east-2/amazon-ssm-agent-updater/2.3.842.0/amazon-ssm-agent-updater-
snap-amd64.tar.gz\nSuccessfully downloaded https://s3.us-east-2.amazonaws.com/
amazon-ssm-us-east-2/amazon-ssm-agent/2.3.672.0/amazon-ssm-agent-snap-amd64.tar.gz
\nSuccessfully downloaded https://s3.us-east-2.amazonaws.com/amazon-ssm-us-east-2/
amazon-ssm-agent/2.3.842.0/amazon-ssm-agent-snap-amd64.tar.gz\nInitiating amazon-
ssm-agent update to 2.3.842.0\namazon-ssm-agent updated successfully to 2.3.842.0",
                    "StandardOutputUrl": "",
                    "StandardErrorUrl": "",
                    "OutputS3Region": "us-east-2",
                    "OutputS3BucketName": "",
                    "OutputS3KeyPrefix": "8bee3135-398c-4d31-99b6-e42d2EXAMPLE/
i-0471e04240EXAMPLE/awsupdateSsmAgent"
```

```
                }
            ],
            "ServiceRole": "",
            "NotificationConfig": {
                "NotificationArn": "",
                "NotificationEvents": [],
                "NotificationType": ""
            },
            "CloudWatchOutputConfig": {
                "CloudWatchLogGroupName": "",
                "CloudWatchOutputEnabled": false
            }
        }
    ]
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 명령 상태 이해를 참조하세요.

- 자세한 API 내용은 명령 참조ListCommandInvocations의 섹션을 참조하세요. AWS CLI

### `list-commands`

다음 코드 예시에서는 `list-commands`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 특정 명령의 상태를 가져오는 방법

다음 `list-commands` 예제에서는 지정된 명령의 상태를 검색하고 표시합니다.

```
aws ssm list-commands \
    --command-id "0831e1a8-a1ac-4257-a1fd-c831bEXAMPLE"
```

예제 2: 특정 날짜 이후에 요청된 명령의 상태를 가져오는 방법

다음 `list-commands` 예제에서는 지정된 날짜 이후에 요청된 명령의 세부 정보를 검색합니다.

```
aws ssm list-commands \
    --filter "key=InvokedAfter,value=2020-02-01T00:00:00Z"
```

예제 3: AWS 계정에서 요청된 모든 명령을 나열하려면

다음 list-commands 예제에서는 현재 AWS 계정 및 리전의 사용자가 요청한 모든 명령을 나열합니다.

```
aws ssm list-commands
```

출력:

```
{
    "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
            }
        }
    ]
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 Systems Manager Run Command를 사용하여 명령 실행을 참조하세요.

- 자세한 API 내용은 명령 참조ListCommands의 섹션을 참조하세요. AWS CLI

### `list-compliance-items`

다음 코드 예시에서는 `list-compliance-items`을 사용하는 방법을 보여 줍니다.

AWS CLI

특정 인스턴스의 규정 준수 항목을 나열하는 방법

이 예제에서는 지정된 인스턴스의 모든 규정 준수 항목을 나열합니다.

명령:

```
aws ssm list-compliance-items --resource-ids "i-1234567890abcdef0" --resource-types "ManagedInstance"
```

출력:

```
{
  "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--"
```

```
}
```

특정 인스턴스 및 연결 ID에 대한 규정 준수 항목을 나열하는 방법

이 예제에서는 지정된 인스턴스 및 연결 ID의 모든 규정 준수 항목을 나열합니다.

명령:

```
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"
```

특정 날짜 및 시간 이후 인스턴스의 규정 준수 항목을 나열하는 방법

이 예제에서는 지정된 날짜 및 시간 이후 인스턴스에 대한 모든 규정 준수 항목을 나열합니다.

명령:

```
aws ssm list-compliance-items --resource-ids "i-1234567890abcdef0" --resource-
types "ManagedInstance" --
filters "Key=ExecutionTime,Values=2019-02-18T16:00:00Z,Type=GREATER_THAN"
```

- 자세한 API 내용은 명령 참조ListComplianceItems의 섹션을 참조하세요. AWS CLI

### `list-compliance-summaries`

다음 코드 예시에서는 `list-compliance-summaries`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 규정 준수 유형에 대한 규정 준수 요약을 나열하는 방법

이 예제에서는 계정의 모든 규정 준수 유형에 대한 규정 준수 요약을 나열합니다.

명령:

```
aws ssm list-compliance-summaries
```

출력:

```
{
  "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": "eyJOZXh0VG9rZW4iOiBudWxsLCAiYm90b190cnVuY2F0ZV9hbW91bnQiOiAyfQ=="
}
```

특정 규정 준수 유형에 대한 규정 준수 요약을 나열하는 방법

이 예제에서는 패치 규정 준수 유형에 대한 규정 준수 요약을 나열합니다.

명령:

```
aws ssm list-compliance-summaries --
filters "Key=ComplianceType,Values=Patch,Type=EQUAL"
```

- 자세한 API 내용은 명령 참조ListComplianceSummaries의 섹션을 참조하세요. AWS CLI

### `list-document-metadata-history`

다음 코드 예시에서는 `list-document-metadata-history`을 사용하는 방법을 보여 줍니다.

AWS CLI

예: 변경 템플릿의 승인 기록 및 상태를 보려면

다음 `list-document-metadata-history` 예제에서는 지정된 Change Manager 변경 템플릿에 대한 승인 기록을 반환합니다.

```
aws ssm list-document-metadata-history \
    --name MyChangeManageTemplate \
    --metadata DocumentReviews
```

출력:

```
{
    "Name": "MyChangeManagerTemplate",
```

```
    "DocumentVersion": "1",
    "Author": "arn:aws:iam::111222333444;:user/JohnDoe",
    "Metadata": {
        "ReviewerResponse": [
            {
                "CreateTime": "2021-07-30T11:58:28.025000-07:00",
                "UpdatedTime": "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",
                "UpdatedTime": "2021-07-30T11:58:28.025000-07:00",
                "ReviewStatus": "PENDING"
            }
        ]
    }
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 변경 템플릿 검토 및 승인 또는 거부를 참조하세요.

- 자세한 API 내용은 명령 참조ListDocumentMetadataHistory의 섹션을 참조하세요. AWS CLI

### `list-document-versions`

다음 코드 예시에서는 `list-document-versions`을 사용하는 방법을 보여 줍니다.

AWS CLI

문서 버전을 나열하는 방법

다음 `list-document-versions` 예제에서는 Systems Manager 문서의 모든 버전을 나열합니다.

```
aws ssm list-document-versions \
    --name "Example"
```

출력:

```
{
    "DocumentVersions": [
        {
            "Name": "Example",
            "DocumentVersion": "1",
            "CreatedDate": 1583257938.266,
            "IsDefaultVersion": true,
            "DocumentFormat": "YAML",
            "Status": "Active"
        }
    ]
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 문서 버전 파라미터를 사용하는 명령 전송을 참조하세요.

- 자세한 API 내용은 명령 참조ListDocumentVersions의 섹션을 참조하세요. AWS CLI

### `list-documents`

다음 코드 예시에서는 `list-documents`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 문서를 나열하는 방법

다음 `list-documents` 예제에서는 사용자 지정 태그로 지정된 요청 계정에서 소유한 문서를 나열합니다.

```
aws ssm list-documents \
    --filters Key=Owner,Values=Self Key=tag:DocUse,Values=Testing
```

출력:

```
{
    "DocumentIdentifiers": [
        {
            "Name": "Example",
            "Owner": "29884EXAMPLE",
```

```
            "PlatformTypes": [
                "Windows",
                "Linux"
            ],
            "DocumentVersion": "1",
            "DocumentType": "Automation",
            "SchemaVersion": "0.3",
            "DocumentFormat": "YAML",
            "Tags": [
                {
                    "Key": "DocUse",
                    "Value": "Testing"
                }
            ]
        }
    ]
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 AWS Systems Manager 문서를 참조하세요.

예제 2: 공유 문서를 나열하는 방법

다음 `list-documents` 예제에서는 에서 소유하지 않은 프라이빗 공유 문서를 포함하여 공유 문서를 나열합니다 AWS.

```
aws ssm list-documents \
    --filters Key=Name,Values=sharedDocNamePrefix  Key=Owner,Values=Private
```

출력:

```
{
    "DocumentIdentifiers": [
        {
            "Name": "Example",
            "Owner": "12345EXAMPLE",
            "PlatformTypes": [
                "Windows",
                "Linux"
            ],
            "DocumentVersion": "1",
            "DocumentType": "Command",
            "SchemaVersion": "0.3",
            "DocumentFormat": "YAML",
```

```
            "Tags": []
        }
    ]
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 AWS Systems Manager 문서를 참조하세요.

- 자세한 API 내용은 명령 참조ListDocuments의 섹션을 참조하세요. AWS CLI

### `list-inventory-entries`

다음 코드 예시에서는 `list-inventory-entries`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 인스턴스의 특정 인벤토리 유형 항목을 보는 방법

다음 `list-inventory-entries` 예제에서는 특정 인스턴스의 AWS:Application Inventory Type에 대한 인벤토리 항목을 나열합니다.

```
aws ssm list-inventory-entries \
    --instance-id "i-1234567890abcdef0" \
    --type-name "AWS:Application"
```

출력:

```
{
  "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"
    }
  ]
}
```

예제 2: 인스턴스에 할당된 사용자 지정 인벤토리 항목을 보는 방법

다음 `list-inventory-entries` 예제에서는 인스턴스에 할당된 사용자 지정 인벤토리 항목을 나열합니다.

```
aws ssm list-inventory-entries \
    --instance-id "i-1234567890abcdef0" \
    --type-name "Custom:RackInfo"
```

출력:

```
{
  "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"
    }
  ]
}
```

- 자세한 API 내용은 명령 참조ListInventoryEntries의 섹션을 참조하세요. AWS CLI

### `list-ops-item-related-items`

다음 코드 예시에서는 `list-ops-item-related-items`을 사용하는 방법을 보여 줍니다.

AWS CLI

의 관련 항목 리소스를 나열하려면 OpsItem

다음 `list-ops-item-related-items` 예제에서는 의 관련 항목 리소스를 나열합니다 OpsItem.

```
aws ssm list-ops-item-related-items \
    --ops-item-id "oi-f99f2EXAMPLE"
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 에서 Incident Manager 인시던트 작업을 OpsCenter 참조하세요.

- 자세한 API 내용은 명령 참조ListOpsItemRelatedItems의 섹션을 참조하세요. AWS CLI

### `list-resource-compliance-summaries`

다음 코드 예시에서는 `list-resource-compliance-summaries`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 수준 규정 준수 요약 수를 나열하는 방법

이 예제에서는 리소스 수준 규정 준수 요약 수를 나열합니다.

명령:

```
aws ssm list-resource-compliance-summaries
```

출력:

```
{
  "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--"
}
```

특정 규정 준수 유형에 대한 리소스 수준 규정 준수 요약을 나열하는 방법

이 예제에서는 패치 규정 준수 유형에 대한 리소스 수준 규정 준수 요약을 나열합니다.

명령:

```
aws ssm list-resource-compliance-summaries --
filters "Key=ComplianceType,Values=Patch,Type=EQUAL"
```

- 자세한 API 내용은 명령 참조ListResourceComplianceSummaries의 섹션을 참조하세요. AWS CLI

### `list-resource-data-sync`

다음 코드 예시에서는 `list-resource-data-sync`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 데이터 동기화 구성을 나열하려면

이 예제에서는 리소스 데이터 동기화 구성에 대한 정보를 검색합니다.

```
aws ssm list-resource-data-sync
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListResourceDataSync의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

패치 기준에 적용된 태그를 나열하는 방법

다음 `list-tags-for-resource` 예제에서는 패치 기준의 태그를 나열합니다.

```
aws ssm list-tags-for-resource \
    --resource-type "PatchBaseline" \
    --resource-id "pb-0123456789abcdef0"
```

출력:

```
{
    "TagList": [
        {
            "Key": "Environment",
            "Value": "Production"
        },
        {
            "Key": "Region",
            "Value": "EMEA"
        }
    ]
}
```

자세한 내용은 AWS 일반 참조의 AWS 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### `modify-document-permission`

다음 코드 예시에서는 `modify-document-permission`을 사용하는 방법을 보여 줍니다.

AWS CLI

문서 권한을 수정하는 방법

다음 `modify-document-permission` 예제에서는 Systems Manager 문서를 공개적으로 공유합니다.

```
aws ssm modify-document-permission \
```

```
    --name "Example" \
    --permission-type "Share" \
    --account-ids-to-add "All"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Systems Manager 사용 설명서의 Systems Manager 문서 공유를 참조하세요.

- 자세한 API 내용은 명령 참조ModifyDocumentPermission의 섹션을 참조하세요. AWS CLI

### put-compliance-items

다음 코드 예시에서는 put-compliance-items을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 인스턴스에 규정 준수 유형 및 규정 준수 세부 정보를 등록하는 방법

이 예제에서는 지정된 관리형 인스턴스에 규정 준수 유형 Custom:AVCheck를 등록합니다. 명령이 성공해도 출력은 없습니다.

명령:

```
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"
```

- 자세한 API 내용은 명령 참조PutComplianceItems의 섹션을 참조하세요. AWS CLI

### put-inventory

다음 코드 예시에서는 put-inventory을 사용하는 방법을 보여 줍니다.

AWS CLI

인스턴스에 사용자 지정 메타데이터를 할당하는 방법

이번 예에서는 인스턴스에 랙 위치 정보를 할당합니다. 명령이 성공해도 출력은 없습니다.

명령(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"}]}]'
```

명령(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'}]"
```

- 자세한 API 내용은 명령 참조PutInventory의 섹션을 참조하세요. AWS CLI

### put-parameter

다음 코드 예시에서는 put-parameter을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: 파라미터 값을 변경하는 방법

다음 put-parameter 예시에서는 지정된 파라미터의 값을 변경합니다.

```
aws ssm put-parameter \
    --name "MyStringParameter" \
    --type "String" \
    --value "Vici" \
    --overwrite
```

출력:

```
{
    "Version": 2,
    "Tier": "Standard"
}
```

자세한 내용은 Systems Manager 사용 설명서의 Systems Manager 파라미터 생성(AWS CLI), '파라미터 티어 관리 <https://docs.aws.amazon.com/systems-manager/latest/userguide/parameter-store-advanced-parameters.html>`__ 및 파라미터 정책 작업을 참조하세요. AWS

예 2: 고급 파라미터를 생성하는 방법

다음 put-parameter 예시에서는 고급 파라미터를 생성합니다.

```
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
```

출력:

```
{
    "Version": 1,
    "Tier": "Advanced"
}
```

자세한 내용은 Systems Manager 사용 설명서의 Systems Manager 파라미터 생성(AWS CLI), '파라미터 티어 관리 <https://docs.aws.amazon.com/systems-manager/latest/userguide/parameter-store-advanced-parameters.html>`__ 및 파라미터 정책 작업을 참조하세요. AWS

예 3: 표준 파라미터를 고급 파라미터로 변환하는 방법

다음 put-parameter 예시에서는 기존 표준 파라미터를 고급 파라미터로 변환합니다.

```
aws ssm put-parameter \
    --name "MyConvertedParameter" \
    --value "abc123" \
    --type "String" \
    --tier Advanced \
    --overwrite
```

출력:

```
{
    "Version": 2,
    "Tier": "Advanced"
```

```
}
```

자세한 내용은 Systems Manager 사용 설명서의 Systems Manager 파라미터 생성(AWS CLI), '파라미터 티어 관리 <https://docs.aws.amazon.com/systems-manager/latest/userguide/parameter-store-advanced-parameters.html>`__ 및 파라미터 정책 작업을 참조하세요. AWS

예 4: 정책이 연결된 파라미터를 생성하는 방법

다음 put-parameter 예시에서는 파라미터 정책이 연결된 고급 파라미터를 생성합니다.

```
aws ssm put-parameter \
    --name "/Finance/Payroll/q2accesskey" \
    --value "P@sSwW)rd" \
    --type "SecureString" \
    --tier Advanced \
    --policies "[{\"Type\":\"Expiration\",\"Version\":\"1.0\",\"Attributes\":
{\"Timestamp\":\"2020-06-30T00:00:00.000Z\"}},{\"Type\":\"ExpirationNotification\",
\"Version\":\"1.0\",\"Attributes\":{\"Before\":\"5\",\"Unit\":\"Days\"}},{\"Type\":
\"NoChangeNotification\",\"Version\":\"1.0\",\"Attributes\":{\"After\":\"60\",\"Unit
\":\"Days\"}}]"
```

출력:

```
{
    "Version": 1,
    "Tier": "Advanced"
}
```

자세한 내용은 Systems Manager 사용 설명서의 Systems Manager 파라미터 생성(AWS CLI), '파라미터 티어 관리 <https://docs.aws.amazon.com/systems-manager/latest/userguide/parameter-store-advanced-parameters.html>`__ 및 파라미터 정책 작업을 참조하세요. AWS

예 5: 기존 파라미터에 정책을 추가하는 방법

다음 put-parameter 예시에서는 정책을 기존 고급 파라미터에 연결합니다.

```
aws ssm put-parameter \
    --name "/Finance/Payroll/q2accesskey" \
    --value "N3wP@sSwW)rd" \
    --type "SecureString" \
```

```
    --tier Advanced \
    --policies "[{\"Type\":\"Expiration\",\"Version\":\"1.0\",\"Attributes\":
{\"Timestamp\":\"2020-06-30T00:00:00.000Z\"}},{\"Type\":\"ExpirationNotification\",
\"Version\":\"1.0\",\"Attributes\":{\"Before\":\"5\",\"Unit\":\"Days\"}},{\"Type\":
\"NoChangeNotification\",\"Version\":\"1.0\",\"Attributes\":{\"After\":\"60\",\"Unit
\":\"Days\"}}]"
    --overwrite
```

출력:

```
{
    "Version": 2,
    "Tier": "Advanced"
}
```

자세한 내용은 Systems Manager 사용 설명서의 Systems Manager 파라미터 생성(AWS CLI), '파라미터 티어 관리 <https://docs.aws.amazon.com/systems-manager/latest/userguide/parameter-store-advanced-parameters.html>`__ 및 파라미터 정책 작업을 참조하세요. AWS

- 자세한 API 내용은 명령 참조PutParameter의 섹션을 참조하세요. AWS CLI

### register-default-patch-baseline

다음 코드 예시에서는 `register-default-patch-baseline`을 사용하는 방법을 보여 줍니다.

AWS CLI

기본 패치 기준을 설정하는 방법

다음 `register-default-patch-baseline` 예제에서는 지정된 사용자 지정 패치 기준을 지원하는 운영 체제 유형의 기본 패치 기준으로 등록합니다.

```
aws ssm register-default-patch-baseline \
    --baseline-id "pb-abc123cf9bEXAMPLE"
```

출력:

```
{
    "BaselineId":"pb-abc123cf9bEXAMPLE"
}
```

다음 `register-default-patch-baseline` 예제에서는 CentOS에 AWS 대해 에서 제공하는 기본 패치 기준을 기본 패치 기준으로 등록합니다.

```
aws ssm register-default-patch-baseline \
    --baseline-id "arn:aws:ssm:us-east-2:733109147000:patchbaseline/
pb-0574b43a65ea646ed"
```

출력:

```
{
    "BaselineId":"pb-abc123cf9bEXAMPLE"
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 사전 정의된 패치 기준 및 사용자 지정 패치 기준 정보를 참조하세요.

- 자세한 API 내용은 명령 참조RegisterDefaultPatchBaseline의 섹션을 참조하세요. AWS CLI

### register-patch-baseline-for-patch-group

다음 코드 예시에서는 `register-patch-baseline-for-patch-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

패치 그룹에 대해 패치 기준을 등록하는 방법

다음 `register-patch-baseline-for-patch-group` 예제에서는 패치 그룹의 패치 기준을 등록합니다.

```
aws ssm register-patch-baseline-for-patch-group \
    --baseline-id "pb-045f10b4f382baeda" \
    --patch-group "Production"
```

출력:

```
{
    "BaselineId": "pb-045f10b4f382baeda",
    "PatchGroup": "Production"
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 패치 그룹 생성 <https://docs.aws.amazon.com/systems-manager/latest/userguide/sysman-patch-group-tagging.html>__ 및 패치 기준에 패치 그룹 추가를 참조하세요.

- 자세한 API 내용은 명령 참조RegisterPatchBaselineForPatchGroup의 섹션을 참조하세요. AWS CLI

### register-target-with-maintenance-window

다음 코드 예시에서는 register-target-with-maintenance-window을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 유지 관리 기간에 단일 대상을 등록하는 방법

다음 register-target-with-maintenance-window 예제에서는 유지 관리 기간에 인스턴스를 등록합니다.

```
aws ssm register-target-with-maintenance-window \
    --window-id "mw-ab12cd34ef56gh78" \
    --target "Key=InstanceIds,Values=i-0000293ffd8c57862" \
    --owner-information "Single instance" \
    --resource-type "INSTANCE"
```

출력:

```
{
    "WindowTargetId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

예제 2: 인스턴스를 사용하여 유지 관리 기간에 여러 대상 등록 IDs

다음 register-target-with-maintenance-window 예제에서는 인스턴스 를 지정하여 유지 관리 기간에 두 인스턴스를 등록합니다IDs.

```
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"
```

출력:

```
{
    "WindowTargetId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

예제 3: 리소스 태그를 사용하여 유지 관리 기간에 대상을 등록하는 방법

다음 `register-target-with-maintenance-window` 예제에서는 인스턴스에 적용되는 리소스 태그를 지정하여 유지 관리 기간에 인스턴스를 등록합니다.

```
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"
```

출력:

```
{
    "WindowTargetId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

예제 4: 태그 키 그룹을 사용하여 대상을 등록하는 방법

다음 `register-target-with-maintenance-window` 예제에서는 키 값에 상관없이 모두 하나 이상의 태그가 지정된 인스턴스를 등록합니다.

```
aws ssm register-target-with-maintenance-window \
    --window-id "mw-0c50858d01EXAMPLE" \
    --resource-type "INSTANCE" \
    --target "Key=tag-key,Values=Name,Instance-Type,CostCenter"
```

출력:

```
{
    "WindowTargetId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

예제 5: 리소스 그룹 이름을 사용하여 대상을 등록하는 방법

다음 register-target-with-maintenance-window 예제에서는 포함된 리소스 유형에 상관없이 지정된 리소스 그룹을 등록합니다.

```
aws ssm register-target-with-maintenance-window \
    --window-id "mw-0c50858d01EXAMPLE" \
    --resource-type "RESOURCE_GROUP" \
    --target "Key=resource-groups:Name,Values=MyResourceGroup"
```

출력:

```
{
    "WindowTargetId":"1a2b3c4d-1a2b-1a2b-1a2b-1a2b3c4d-1a2"
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 유지 관리 기간(AWS CLI)으로 대상 인스턴스 등록을 참조하세요.

- 자세한 API 내용은 명령 참조RegisterTargetWithMaintenanceWindow의 섹션을 참조하세요. AWS CLI

### register-task-with-maintenance-window

다음 코드 예시에서는 register-task-with-maintenance-window을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 유지 관리 기간에 자동화 작업을 등록하는 방법

다음 register-task-with-maintenance-window 예제에서는 인스턴스에서 대상으로 지정된 유지 관리 기간에 자동화 작업을 등록합니다.

```
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"
```

출력:

```
{
    "WindowTaskId":"11144444-5555-6666-7777-88888888"
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 유지 관리 기간(AWS CLI)으로 작업 등록을 참조하세요.

예제 2: 유지 관리 기간에 Lambda 작업을 등록하는 방법

다음 `register-task-with-maintenance-window` 예제에서는 인스턴스에서 대상으로 지정된 유지 관리 기간에 Lambda 작업을 등록합니다.

```
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"
```

출력:

```
{
    "WindowTaskId":"22244444-5555-6666-7777-88888888"
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 유지 관리 기간(AWS CLI)으로 작업 등록을 참조하세요.

예제 3: 유지 관리 기간에 Run Command 작업을 등록하는 방법

다음 register-task-with-maintenance-window 예제에서는 인스턴스에서 대상으로 지정된 유지 관리 기간에 Run Command 작업을 등록합니다.

```
aws ssm register-task-with-maintenance-window \
    --window-id "mw-082dcd7649dee04e4" \
    --targets "Key=InstanceIds,Values=i-12344d305eEXAMPLE" \
    --service-role-arn "arn:aws:iam::111222333444:role/SSM" \
    --task-type "RUN_COMMAND" \
    --name "SSMInstallPowerShellModule" \
    --task-arn "AWS-InstallPowerShellModule" \
    --task-invocation-parameters "{\"RunCommand\":{\"Comment\":\"\",
\"OutputS3BucketName\":\"runcommandlogs\",\"Parameters\":{\"commands\":[\"Get-
Module -ListAvailable\"],\"executionTimeout\":[\"3600\"],\"source\":[\"https:\/
\/gallery.technet.microsoft.com\/EZOut-33ae0fb7\/file\/110351\/1\/EZOut.zip\"],
\"workingDirectory\":[\"\\\\\"]},\"TimeoutSeconds\":600}}" \
    --max-concurrency 1 \
    --max-errors 1 \
    --priority 10
```

출력:

```
{
    "WindowTaskId":"33344444-5555-6666-7777-88888888"
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 유지 관리 기간(AWS CLI)으로 작업 등록을 참조하세요.

예제 4: 유지 관리 기간에 Step Functions 작업을 등록하는 방법

다음 register-task-with-maintenance-window 예제에서는 인스턴스에서 대상으로 지정된 유지 관리 기간에 Step Functions 작업을 등록합니다.

```
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"
```

출력:

```
{
    "WindowTaskId":"44444444-5555-6666-7777-88888888"
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 유지 관리 기간(AWS CLI)으로 작업 등록을 참조하세요.

예제 5: 유지 관리 기간 대상 ID를 사용하여 작업을 등록하는 방법

다음 `register-task-with-maintenance-window` 예제에서는 유지 관리 기간 대상 ID를 사용하여 작업을 등록합니다. 유지 관리 기간 대상 ID는 `aws ssm register-target-with-maintenance-window` 명령 출력에 포함되어 있습니다. `aws ssm describe-maintenance-window-targets` 명령의 출력에서 검색할 수도 있습니다.

```
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
```

출력:

```
{
    "WindowTaskId":"33344444-5555-6666-7777-88888888"
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 유지 관리 기간(AWS CLI)으로 작업 등록을 참조하세요.

- 자세한 API 내용은 명령 참조RegisterTaskWithMaintenanceWindow의 섹션을 참조하세요. AWS CLI

### `remove-tags-from-resource`

다음 코드 예시에서는 `remove-tags-from-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

패치 기준에서 태그를 삭제하는 방법

다음 `remove-tags-from-resource` 예제에서는 패치 기준에서 태그를 제거합니다.

```
aws ssm remove-tags-from-resource \
    --resource-type "PatchBaseline" \
    --resource-id "pb-0123456789abcdef0" \
    --tag-keys "Region"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS 일반 참조의 AWS 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조RemoveTagsFromResource의 섹션을 참조하세요. AWS CLI

### `reset-service-setting`

다음 코드 예시에서는 `reset-service-setting`을 사용하는 방법을 보여 줍니다.

AWS CLI

Parameter Store 처리량에 대한 서비스 설정을 재설정하려면

다음 `reset-service-setting` 예제에서는 지정된 리전의 Parameter Store 처리량에 대한 서비스 설정을 재설정하여 더 이상 처리량 증가를 사용하지 않도록 합니다.

```
aws ssm reset-service-setting \
    --setting-id arn:aws:ssm:us-east-1:123456789012:servicesetting/ssm/parameter-store/high-throughput-enabled
```

출력:

```
{
```

```
    "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"
    }
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 파라미터 스토어 처리량 증가를 참조하세요.

- 자세한 API 내용은 명령 참조ResetServiceSetting의 섹션을 참조하세요. AWS CLI

### `resume-session`

다음 코드 예시에서는 `resume-session`을 사용하는 방법을 보여 줍니다.

AWS CLI

Session Manager 세션을 재개하려면

이 `resume-session` 예제에서는 인스턴스 연결이 끊긴 후 인스턴스와 함께 Session Manager 세션을 재개합니다. 참고로 이 대화형 명령을 사용하려면 호출을 수행하는 클라이언트 시스템에 Session Manager 플러그인을 설치해야 합니다.

```
aws ssm resume-session \
    --session-id Mary-Major-07a16060613c408b5
```

출력:

```
{
    "SessionId": "Mary-Major-07a16060613c408b5",
    "TokenValue":
 "AAEAAVbTGsaOnyvcUoNGqifbv5r/8lgxuQljCuY8qVcvOnoBAAAAAFxtd3jIXAFUUXGTJ7zF/
AWJPwDviOlF5p3dlAgrqVIVO6IEXhkHLz0/1gXKRKEME71E6TLOplLDJAMZ
+kREejkZu4c5AxMkrQjMF+gtHP1bYJKTwtHQd1wjulPLexO8SHl7g5R/
wekrj6WsDUpnEegFBfGftpAIz2GXQVfTJXKfkc5qepQ11C11DOIT2dozOqXgHwfQHfAKLErM5dWDZqKwyT1Z3iw7unQd
+ihfGa6MEJJ97Jmat/a2TspEnOjNn9Mvu5iwXIW2yCvWZrGUj+/
QI5Xr7s1XJBEnSKR54o4fN0GV9RWl0RZsZm1m1ki0JJtiwwgZ",
    "StreamUrl": "wss://ssmmessages.us-east-2.amazonaws.com/v1/data-channel/Mary-
Major-07a16060613c408b5?role=publish_subscribe"
```

```
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 에 대한 Session Manager 플러그인 설치를 AWS CLI 참조하세요.

- 자세한 API 내용은 명령 참조ResumeSession의 섹션을 참조하세요. AWS CLI

### send-automation-signal

다음 코드 예시에서는 send-automation-signal을 사용하는 방법을 보여 줍니다.

AWS CLI

자동화 실행에 신호를 보내려면

다음 send-automation-signal 예제에서는 승인 신호를 자동화 실행으로 보냅니다.

```
aws ssm send-automation-signal \
    --automation-execution-id 73c8eef8-f4ee-4a05-820c-e354fEXAMPLE \
    --signal-type "Approve"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Systems Manager 사용 설명서의 승인자와 함께 자동화 워크플로 실행을 참조하세요.

- 자세한 API 내용은 명령 참조SendAutomationSignal의 섹션을 참조하세요. AWS CLI

### send-command

다음 코드 예시에서는 send-command을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 하나 이상의 원격 인스턴스에서 명령을 실행하는 방법

다음 send-command 예제에서는 대상 인스턴스에서 echo 명령을 실행합니다.

```
aws ssm send-command \
    --document-name "AWS-RunShellScript" \
    --parameters 'commands=["echo HelloWorld"]' \
    --targets "Key=instanceids,Values=i-1234567890abcdef0" \
```

```
    --comment "echo HelloWorld"
```

출력:

```
{
    "Command": {
        "CommandId": "92853adf-ba41-4cd6-9a88-142d1EXAMPLE",
        "DocumentName": "AWS-RunShellScript",
        "DocumentVersion": "",
        "Comment": "echo HelloWorld",
        "ExpiresAfter": 1550181014.717,
        "Parameters": {
            "commands": [
                "echo HelloWorld"
            ]
        },
        "InstanceIds": [
            "i-0f00f008a2dcbefe2"
        ],
        "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
        }
    }
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 Systems Manager Run Command를 사용하여 명령 실행을 참조하세요.

예제 2: 인스턴스에 대한 IP 정보를 가져오는 방법

다음 send-command 예제에서는 인스턴스에 대한 IP 정보를 검색합니다.

```
aws ssm send-command \
    --instance-ids "i-1234567890abcdef0" \
    --document-name "AWS-RunShellScript" \
    --comment "IP config" \
    --parameters "commands=ifconfig"
```

샘플 출력은 예 1을 참조하세요.

자세한 내용은 AWS Systems Manager 사용 설명서의 Systems Manager Run Command를 사용하여 명령 실행을 참조하세요.

예제 3: 특정 태그를 사용하는 인스턴스에서 명령을 실행하는 방법

다음 send-command 예제에서는 태그 키가 “ENV”이고 값이 “Dev”인 인스턴스에서 명령을 실행합니다.

```
aws ssm send-command \
    --targets "Key=tag:ENV,Values=Dev" \
    --document-name "AWS-RunShellScript" \
    --parameters "commands=ifconfig"
```

샘플 출력은 예 1을 참조하세요.

자세한 내용은 AWS Systems Manager 사용 설명서의 Systems Manager Run Command를 사용하여 명령 실행을 참조하세요.

예제 4: SNS 알림을 보내는 명령을 실행하려면

다음 send-command 예제에서는 모든 SNS 알림 이벤트와 알림 유형에 대한 Command 알림을 보내는 명령을 실행합니다.

```
aws ssm send-command \
    --instance-ids "i-1234567890abcdef0" \
    --document-name "AWS-RunShellScript" \
```

```
    --comment "IP config" \
    --parameters "commands=ifconfig" \
    --service-role-arn "arn:aws:iam::123456789012:role/SNS_Role" \
    --notification-config "NotificationArn=arn:aws:sns:us-
east-1:123456789012:SNSTopicName,NotificationEvents=All,NotificationType=Command"
```

샘플 출력은 예 1을 참조하세요.

자세한 내용은 AWS Systems Manager 사용 설명서의 Systems Manager Run Command를 사용하여 명령 실행을 참조하세요.

예제 5: S3 및 에 출력하는 명령을 실행하려면 CloudWatch

다음 send-command 예제에서는 명령 세부 정보를 S3 버킷 및 CloudWatch Logs 로그 그룹에 출력하는 명령을 실행합니다.

```
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"
```

샘플 출력은 예 1을 참조하세요.

자세한 내용은 AWS Systems Manager 사용 설명서의 Systems Manager Run Command를 사용하여 명령 실행을 참조하세요.

예제 6: 태그가 서로 다른 여러 인스턴스에서 명령을 실행하는 방법

다음 send-command 예제는 서로 다른 두 개의 태그 키와 값을 가진 인스턴스에서 명령을 실행합니다.

```
aws ssm send-command \
    --document-name "AWS-RunPowerShellScript" \
    --parameters commands=["echo helloWorld"] \
    --targets Key=tag:Env,Values=Dev Key=tag:Role,Values=WebServers
```

샘플 출력은 예 1을 참조하세요.

자세한 내용은 AWS Systems Manager 사용 설명서의 Systems Manager Run Command를 사용하여 명령 실행을 참조하세요.

예제 7: 태그 키가 같은 여러 인스턴스를 대상으로 지정하는 방법

다음 send-command 예제에서는 태그 키는 같지만 값이 다른 인스턴스에서 명령을 실행합니다.

```
aws ssm send-command \
    --document-name "AWS-RunPowerShellScript" \
    --parameters commands=["echo helloWorld"] \
    --targets Key=tag:Env,Values=Dev,Test
```

샘플 출력은 예 1을 참조하세요.

자세한 내용은 AWS Systems Manager 사용 설명서의 Systems Manager Run Command를 사용하여 명령 실행을 참조하세요.

예제 8: 공유 문서를 사용하는 명령을 실행하는 방법

다음 send-command 예제에서는 대상 인스턴스에서 공유 문서를 실행합니다.

```
aws ssm send-command \
    --document-name "arn:aws:ssm:us-east-1:123456789012:document/ExampleDocument" \
    --targets "Key=instanceids,Values=i-1234567890abcdef0"
```

샘플 출력은 예 1을 참조하세요.

자세한 내용은 AWS Systems Manager 사용 설명서의 공유 SSM 문서 사용을 참조하세요.

- 자세한 API 내용은 명령 참조SendCommand의 섹션을 참조하세요. AWS CLI

### start-associations-once

다음 코드 예시에서는 start-associations-once을 사용하는 방법을 보여 줍니다.

AWS CLI

연결을 즉시 한 번만 실행하려면

다음 start-associations-once 예제에서는 지정된 연결을 즉시 한 번만 실행합니다. 명령이 성공해도 출력은 없습니다.

```
aws ssm start-associations-once \
```

```
    --association-id "8dfe3659-4309-493a-8755-0123456789ab"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Systems Manager 사용 설명서의 연결 기록 보기를 참조하세요.

- 자세한 API 내용은 명령 참조StartAssociationsOnce의 섹션을 참조하세요. AWS CLI

### start-automation-execution

다음 코드 예시에서는 start-automation-execution을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 자동화 문서를 실행하는 방법

다음 start-automation-execution 예제에서는 자동화 문서를 실행합니다.

```
aws ssm start-automation-execution \
    --document-name "AWS-UpdateLinuxAmi" \
    --parameters "AutomationAssumeRole=arn:aws:iam::123456789012:role/
SSMAutomationRole,SourceAmiId=ami-EXAMPLE,IamInstanceProfileName=EC2InstanceRole"
```

출력:

```
{
  "AutomationExecutionId": "4105a4fc-f944-11e6-9d32-0a1b2EXAMPLE"
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 수동으로 자동화 워크플로 실행을 참조하세요.

예제 2: 공유 자동화 문서를 실행하는 방법

다음 start-automation-execution 예제에서는 공유 자동화 문서를 실행합니다.

```
aws ssm start-automation-execution \
    --document-name "arn:aws:ssm:us-east-1:123456789012:document/ExampleDocument"
```

출력:

```
{
```

```
  "AutomationExecutionId": "4105a4fc-f944-11e6-9d32-0a1b2EXAMPLE"
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 공유 SSM 문서 사용을 참조하세요.

- 자세한 API 내용은 명령 참조StartAutomationExecution의 섹션을 참조하세요. AWS CLI

### `start-change-request-execution`

다음 코드 예시에서는 `start-change-request-execution`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 변경 요청을 시작하려면

다음 `start-change-request-execution` 예제에서는 지정된 최소 옵션으로 변경 요청을 시작합니다.

```
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"
```

출력:

```
{
  "AutomationExecutionId": "9d32a4fc-f944-11e6-4105-0a1b2EXAMPLE"
}
```

예제 2: 외부 JSON 파일을 사용하여 변경 요청을 시작하려면

다음 `start-automation-execution` 예제에서는 JSON 파일에 지정된 여러 옵션을 사용하여 변경 요청을 시작합니다.

```
aws ssm start-change-request-execution \
    --cli-input-json file://MyChangeRequest.json
```

MyChangeRequest.json의 콘텐츠:

```
{
    "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"
}
```

출력:

```
{
  "AutomationExecutionId": "9d32a4fc-f944-11e6-4105-0a1b2EXAMPLE"
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 변경 요청 생성을 참조하세요.

- 자세한 API 내용은 명령 참조StartChangeRequestExecution의 섹션을 참조하세요. AWS CLI

### `start-session`

다음 코드 예시에서는 `start-session`을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: Session Manager 세션 시작

이 `start-session` 예제는 Session Manager 세션을 위해 인스턴스에 대한 연결을 설정합니다. 참고로 이 대화형 명령을 사용하려면 호출을 수행하는 클라이언트 시스템에 Session Manager 플러그인을 설치해야 합니다.

```
aws ssm start-session \
    --target "i-1234567890abcdef0"
```

출력:

```
Starting session with SessionId: Jane-Roe-07a16060613c408b5
```

예제 2: 를 사용하여 Session Manager 세션을 시작하려면 SSH

이 `start-session` 예제에서는 를 사용하여 Session Manager 세션의 인스턴스와 연결합니다 SSH. 이 대화형 명령을 사용하려면 호출을 수행하는 클라이언트 시스템에 Session Manager 플러그인을 설치해야 하며, Linux용 인스턴스와 같은 `ec2-user` EC2 인스턴스의 기본 사용자를 명령에 사용해야 합니다.

```
ssh -i /path/my-key-pair.pem ec2-user@i-02573cafcfEXAMPLE
```

출력:

```
Starting session with SessionId: ec2-user-07a16060613c408b5
```

자세한 내용은 AWS Systems Manager 사용 설명서의 세션 시작 및 세션 관리자 플러그인 설치를 참조하세요. AWS CLI

- 자세한 API 내용은 명령 참조StartSession의 섹션을 참조하세요. AWS CLI

### stop-automation-execution

다음 코드 예시에서는 stop-automation-execution을 사용하는 방법을 보여 줍니다.

AWS CLI

자동화 실행을 중지하는 방법

다음 stop-automation-execution 예제에서는 자동화 문서를 중지합니다.

```
aws ssm stop-automation-execution
    --automation-execution-id "4105a4fc-f944-11e6-9d32-0a1b2EXAMPLE"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Systems Manager 사용 설명서의 수동으로 자동화 워크플로 실행을 참조하세요.

- 자세한 API 내용은 명령 참조StopAutomationExecution의 섹션을 참조하세요. AWS CLI

### terminate-session

다음 코드 예시에서는 terminate-session을 사용하는 방법을 보여 줍니다.

AWS CLI

Session Manager 세션을 종료하려면

이 terminate-session 예제는 사용자가 생성한 세션인 “Shirley-Rodriguez”를 영구적으로 종료하고 Session Manager 클라이언트와 인스턴스의 SSM 에이전트 간의 데이터 연결을 닫습니다.

```
aws ssm terminate-session \
    --session-id "Shirley-Rodriguez-07a16060613c408b5"
```

출력:

```
{
    "SessionId": "Shirley-Rodriguez-07a16060613c408b5"
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 세션 종료를 참조하세요.

- 자세한 API 내용은 명령 참조TerminateSession의 섹션을 참조하세요. AWS CLI

### unlabel-parameter-version

다음 코드 예시에서는 unlabel-parameter-version을 사용하는 방법을 보여 줍니다.

AWS CLI

파라미터 레이블을 삭제하려면

다음 unlabel-parameter-version 예제에서는 지정된 파라미터 버전에서 지정된 레이블을 삭제합니다.

```
aws ssm unlabel-parameter-version \
    --name "parameterName" \
    --parameter-version "version" \
    --labels "label_1" "label_2" "label_3"
```

출력:

```
{
    "RemovedLabels": [
        "label_1"
        "label_2"
        "label_3"
    ],
    "InvalidLabels": []
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 파라미터 레이블 삭제(AWS CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조UnlabelParameterVersion의 섹션을 참조하세요. AWS CLI

### update-association-status

다음 코드 예시에서는 update-association-status을 사용하는 방법을 보여 줍니다.

AWS CLI

연결 상태를 업데이트하는 방법

다음 update-association-status 예제에서는 인스턴스와 문서 간 연결의 연결 상태를 업데이트합니다.

```
aws ssm update-association-status \
    --name "AWS-UpdateSSMAgent" \
    --instance-id "i-1234567890abcdef0" \
    --association-
status "Date=1424421071.939,Name=Pending,Message=temp_status_change,AdditionalInfo=Additiona
Config-Needed"
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 AWS  Systems Manager 사용 설명서의 Systems Manager에서 연결 작업을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateAssociationStatus의 섹션을 참조하세요. AWS CLI

### update-association

다음 코드 예시에서는 update-association을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 문서 연결을 업데이트하는 방법

다음 update-association 예제에서는 새 문서 버전과의 연결을 업데이트합니다.

```
aws ssm update-association \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab" \
    --document-version "\$LATEST"
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 새 연결 버전 편집 및 생성을 참조하세요.

예제 2: 연결의 일정 표현식을 업데이트하는 방법

다음 update-association 예제에서는 지정된 연결의 일정 표현식을 업데이트합니다.

```
aws ssm update-association \
    --association-id "8dfe3659-4309-493a-8755-0123456789ab" \
    --schedule-expression "cron(0 0 0/4 1/1 * ? *)"
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 새 연결 버전 편집 및 생성을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateAssociation의 섹션을 참조하세요. AWS CLI

### update-document-default-version

다음 코드 예시에서는 update-document-default-version을 사용하는 방법을 보여 줍니다.

AWS CLI

문서의 기본 버전을 업데이트하는 방법

다음 update-document-default-version 예제에서는 Systems Manager 문서의 기본 버전을 업데이트합니다.

```
aws ssm update-document-default-version \
    --name "Example" \
    --document-version "2"
```

출력:

```
{
    "Description": {
        "Name": "Example",
        "DefaultVersion": "2"
    }
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 SSM 문서 콘텐츠 작성을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateDocumentDefaultVersion의 섹션을 참조하세요. AWS CLI

### update-document-metadata

다음 코드 예시에서는 update-document-metadata을 사용하는 방법을 보여 줍니다.

AWS CLI

예: 변경 템플릿의 최신 버전을 승인하려면

다음은 검토를 위해 제출된 변경 템플릿의 최신 버전에 대한 승인을 update-document-metadata 제공합니다.

```
aws ssm update-document-metadata \
    --name MyChangeManagerTemplate \
    --document-reviews 'Action=Approve,Comment=[{Type=Comment,Content=Approved!}]'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Systems Manager 사용 설명서의 변경 템플릿 검토 및 승인 또는 거부를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateDocumentMetadata의 섹션을 참조하세요. AWS CLI

### update-document

다음 코드 예시에서는 update-document을 사용하는 방법을 보여 줍니다.

AWS CLI

문서의 새 버전을 생성하는 방법

다음 update-document 예제에서는 Windows 컴퓨터에서 실행 시 문서의 새 버전을 생성합니다. 에서 지정한 문서는 JSON 형식이어야 --document 합니다. 콘텐츠 파일 경로 앞에서 file://을 참조해야 합니다. --document-version 파라미터의 시작 위치에 $이 있으므로 Windows에서는 값을 큰따옴표로 묶어야 합니다. Linux, MacOS 또는 PowerShell 프롬프트에서 값을 작은따옴표로 묶어야 합니다.

Windows 버전:

```
aws ssm update-document \
```

```
    --name "RunShellScript" \
    --content "file://RunShellScript.json" \
    --document-version "$LATEST"
```

Linux 및 Mac 버전:

```
aws ssm update-document \
    --name "RunShellScript" \
    --content "file://RunShellScript.json" \
    --document-version '$LATEST'
```

출력:

```
{
  "DocumentDescription": {
      "Status": "Updating",
      "Hash": "f775e5df4904c6fa46686c4722fae9de1950dace25cd9608ff8d622046b68d9b",
      "Name": "RunShellScript",
      "Parameters": [
          {
              "Type": "StringList",
              "Name": "commands",
              "Description": "(Required) Specify a shell script or a command to
 run."
          }
      ],
      "DocumentType": "Command",
      "PlatformTypes": [
          "Linux"
      ],
      "DocumentVersion": "2",
      "HashType": "Sha256",
      "CreatedDate": 1487899655.152,
      "Owner": "809632081692",
      "SchemaVersion": "2.0",
      "DefaultVersion": "1",
      "LatestVersion": "2",
      "Description": "Run an updated script"
  }
}
```

- 자세한 API 내용은 명령 참조UpdateDocument의 섹션을 참조하세요. AWS CLI

### update-maintenance-window-target

다음 코드 예시에서는 update-maintenance-window-target을 사용하는 방법을 보여 줍니다.

AWS CLI

유지 관리 기간 대상을 업데이트하려면

다음 update-maintenance-window-target 예제에서는 유지 관리 기간 대상의 이름만 업데이트합니다.

```
aws ssm update-maintenance-window-target \
    --window-id "mw-0c5ed765acEXAMPLE" \
    --window-target-id "57e8344e-fe64-4023-8191-6bf05EXAMPLE" \
    --name "NewName" \
    --no-replace
```

출력:

```
{
    "Description": "",
    "OwnerInformation": "",
    "WindowTargetId": "57e8344e-fe64-4023-8191-6bf05EXAMPLE",
    "WindowId": "mw-0c5ed765acEXAMPLE",
    "Targets": [
        {
            "Values": [
                "i-1234567890EXAMPLE"
            ],
            "Key": "InstanceIds"
        }
    ],
    "Name": "NewName"
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 유지 관리 기간 업데이트(AWS CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateMaintenanceWindowTarget의 섹션을 참조하세요. AWS CLI

### update-maintenance-window-task

다음 코드 예시에서는 update-maintenance-window-task을 사용하는 방법을 보여 줍니다.

AWS CLI

유지 관리 기간 작업을 업데이트하려면

다음 update-maintenance-window-task 예제에서는 유지 관리 기간 작업에 대한 서비스 역할을 업데이트합니다.

```
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"
```

출력:

```
{
    "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"
}
```

자세한 내용은 AWS Systems Manager 사용 설명서의 유지 관리 기간 업데이트(AWS CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateMaintenanceWindowTask의 섹션을 참조하세요. AWS CLI

### update-maintenance-window

다음 코드 예시에서는 update-maintenance-window을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 유지 관리 기간을 업데이트하는 방법

다음 update-maintenance-window 예제에서는 유지 관리 기간의 이름을 업데이트합니다.

```
aws ssm update-maintenance-window \
    --window-id "mw-1a2b3c4d5e6f7g8h9" \
    --name "My-Renamed-MW"
```

출력:

```
{
    "Cutoff": 1,
    "Name": "My-Renamed-MW",
    "Schedule": "cron(0 16 ? * TUE *)",
    "Enabled": true,
    "AllowUnassociatedTargets": true,
    "WindowId": "mw-1a2b3c4d5e6f7g8h9",
    "Duration": 4
}
```

예제 2: 유지 관리 기간을 비활성화하는 방법

다음 update-maintenance-window 예제에서는 유지 관리 기간을 비활성화합니다.

```
aws ssm update-maintenance-window \
    --window-id "mw-1a2b3c4d5e6f7g8h9" \
    --no-enabled
```

예제 3: 유지 관리 기간을 활성화하는 방법

다음 update-maintenance-window 예제에서는 유지 관리 기간을 활성화합니다.

```
aws ssm update-maintenance-window \
    --window-id "mw-1a2b3c4d5e6f7g8h9" \
    --enabled
```

자세한 내용은 AWS Systems Manager 사용 설명서의 유지 관리 기간 업데이트(AWS CLI)를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateMaintenanceWindow의 섹션을 참조하세요. AWS CLI

### update-managed-instance-role

다음 코드 예시에서는 update-managed-instance-role을 사용하는 방법을 보여 줍니다.

AWS CLI

관리형 인스턴스의 IAM 역할을 업데이트하려면

다음 update-managed-instance-role 예제에서는 관리형 IAM 인스턴스의 인스턴스 프로파일을 업데이트합니다.

```
aws ssm update-managed-instance-role \
    --instance-id "mi-08ab247cdfEXAMPLE" \
    --iam-role "ExampleRole"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Systems Manager 사용 설명서의 4단계: Systems Manager용 IAM 인스턴스 프로파일 생성을 참조하세요. AWS

- 자세한 API 내용은 명령 참조UpdateManagedInstanceRole의 섹션을 참조하세요. AWS CLI

### update-ops-item

다음 코드 예시에서는 update-ops-item을 사용하는 방법을 보여 줍니다.

AWS CLI

를 업데이트하려면 OpsItem

다음 update-ops-item 예제에서는 에 대한 설명, 우선 순위 및 범주를 업데이트합니다 OpsItem. 또한 명령은 OpsItem 편집하거나 변경할 때 알림이 전송되는 SNS 주제를 지정합니다.

```
aws ssm update-ops-item \
    --ops-item-id "oi-287b5EXAMPLE" \
    --description "Primary OpsItem for failover event 2020-01-01-fh398yf" \
    --priority 2 \
    --category "Security" \
    --notifications "Arn=arn:aws:sns:us-east-2:111222333444:my-us-east-2-topic"
```

출력:

```
This command produces no output.
```

자세한 내용은 AWS Systems Manager 사용 설명서 의 작업 OpsItems 단원을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateOpsItem의 섹션을 참조하세요. AWS CLI

### update-patch-baseline

다음 코드 예시에서는 update-patch-baseline을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 패치 기준을 업데이트하는 방법

다음 update-patch-baseline 예제에서는 지정된 두 개의 패치를 거부된 패치로 추가하고 하나의 패치를 기존 패치 기준에 승인된 패치로 추가합니다.

```
aws ssm update-patch-baseline \
        --baseline-id "pb-0123456789abcdef0" \
        --rejected-patches "KB2032276" "MS10-048" \
        --approved-patches "KB2124261"
```

출력:

```
{
    "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",
    "CreatedDate": 1550244180.465,
    "ModifiedDate": 1550244180.465,
    "Description": "Patches for Windows Servers",
    "Sources": []
}
```

예제 2: 패치 기준의 이름을 바꾸는 방법

다음 update-patch-baseline 예제에서는 지정된 패치 기준의 이름을 바꿉니다.

```
aws ssm update-patch-baseline \
    --baseline-id "pb-0713accee01234567" \
    --name "Windows-Server-2012-R2-Important-and-Critical-Security-Updates"
```

자세한 내용은 AWS Systems Manager 사용 설명서의 패치 기준 업데이트 또는 삭제 <https://docs.aws.amazon.com/systems-manager/latest/userguide/patch-baseline-update-or-delete.html>`__를 참조하세요.

- 자세한 API 내용은 명령 참조UpdatePatchBaseline의 섹션을 참조하세요. AWS CLI

### update-resource-data-sync

다음 코드 예시에서는 update-resource-data-sync을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 데이터 동기화를 업데이트하려면

다음 update-resource-data-sync 예제에서는 SyncFromSource 리소스 데이터 동기화를 업데이트합니다.

```
aws ssm update-resource-data-sync \
    --sync-name exampleSync \
    --sync-type SyncFromSource \
    --sync-source '{"SourceType":"SingleAccountMultiRegions", "SourceRegions":["us-east-1", "us-west-2"]}'
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Systems Manager 사용 설명서의 여러 계정 및 리전의 데이터를 표시하도록 Systems Manager 탐색기 설정을 참조하세요. AWS

- 자세한 API 내용은 명령 참조UpdateResourceDataSync의 섹션을 참조하세요. AWS CLI

### update-service-setting

다음 코드 예시에서는 update-service-setting을 사용하는 방법을 보여 줍니다.

AWS CLI

Parameter Store 처리량에 대한 서비스 설정을 업데이트하려면

다음 update-service-setting 예제에서는 처리량 증가를 사용하도록 지정된 리전의 Parameter Store 처리량에 대한 현재 서비스 설정을 업데이트합니다.

```
aws ssm update-service-setting \
```

```
    --setting-id arn:aws:ssm:us-east-1:123456789012:servicesetting/ssm/parameter-
store/high-throughput-enabled \
    --setting-value true
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS Systems Manager 사용 설명서의 파라미터 스토어 처리량 증가를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateServiceSetting의 섹션을 참조하세요. AWS CLI

## 를 사용한 Amazon Textract 예제 AWS CLI

다음 코드 예제에서는 Amazon Textract와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `analyze-document`

다음 코드 예시에서는 `analyze-document`을 사용하는 방법을 보여 줍니다.

AWS CLI

문서의 텍스트를 분석하는 방법

다음 `analyze-document` 예시에서는 문서에서 텍스트를 분석하는 방법을 보여줍니다.

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
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 Amazon Textract 개발자 안내서의 Analyzing Document Text with Amazon Textract 를 참조하세요.

- 자세한 API 내용은 명령 참조AnalyzeDocument의 섹션을 참조하세요. AWS CLI

### detect-document-text

다음 코드 예시에서는 detect-document-text을 사용하는 방법을 보여 줍니다.

AWS CLI

문서에서 텍스트를 감지하는 방법

다음 detect-document-text 예시에서는 문서에서 텍스트를 감지하는 방법을 보여줍니다.

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
```

출력:


```
{
    "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
    }
}
```

자세한 내용은 Amazon Textract 개발자 안내서의 Detecting Document Text with Amazon Textract를 참조하세요.

- 자세한 API 내용은 명령 참조DetectDocumentText의 섹션을 참조하세요. AWS CLI

### get-document-analysis

다음 코드 예시에서는 get-document-analysis을 사용하는 방법을 보여 줍니다.

AWS CLI

여러 페이지 문서의 비동기 텍스트 분석 결과를 가져오는 방법

다음 get-document-analysis 예시에서는 여러 페이지 문서의 비동기 텍스트 분석 결과를 가져오는 방법을 보여줍니다.

```
aws textract get-document-analysis \
    --job-id df7cf32ebbd2a5de113535fcf4d921926a701b09b4e7d089f3aebadb41e0712b \
    --max-results 1000
```

출력:

```
{
    "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"
}
```


자세한 내용은 Amazon Textract 개발자 안내서의 Detecting and Analyzing Text in Multi-Page Documents를 참조하세요.

- 자세한 API 내용은 명령 참조GetDocumentAnalysis의 섹션을 참조하세요. AWS CLI

### `get-document-text-detection`

다음 코드 예시에서는 `get-document-text-detection`을 사용하는 방법을 보여 줍니다.

AWS CLI

다중 페이지 문서에서 비동기 텍스트 감지 결과를 가져오려면

다음 get-document-text-detection 예제는 여러 페이지 문서에서 비동기 텍스트 감지 결과를 가져오는 방법을 보여줍니다.

```
aws textract get-document-text-detection \
    --job-id 57849a3dc627d4df74123dca269d69f7b89329c870c65bb16c9fd63409d200b9 \
    --max-results 1000
```

출력

```
{
    "Blocks": [
        {
            "Geometry": {
                "BoundingBox": {
                    "Width": 1.0,
                    "Top": 0.0,
                    "Left": 0.0,
                    "Height": 1.0
                },
                "Polygon": [
                    {
                        "Y": 0.0,
                        "X": 0.0
                    },
                    {
                        "Y": 0.0,
                        "X": 1.0
                    },
                    {
                        "Y": 1.0,
                        "X": 1.0
                    },
                    {
                        "Y": 1.0,
                        "X": 0.0
                    }
                ]
            },
            "Relationships": [
                {
                    "Type": "CHILD",
                    "Ids": [
                        "1b926a34-0357-407b-ac8f-ec473160c6a9",
```

```
                        "0c35dc17-3605-4c9d-af1a-d9451059df51",
                        "dea3db8a-52c2-41c0-b50c-81f66f4aa758"
                    ]
                }
            ],
            "BlockType": "PAGE",
            "Id": "84671a5e-8c99-43be-a9d1-6838965da33e",
            "Page": 1
        }
    ],
    "NextToken": "GcqyoAJuZwujOT35EN4LCI3EUzMtiLq3nKyFFHvU5q1SaIdEBcSty+njNgoWwuMP/
muqc96S4o5NzDqehhXvhkodMyVO5OJGyms5lsrCxibWJw==",
    "DocumentMetadata": {
        "Pages": 1
    },
    "JobStatus": "SUCCEEDED"
}
```

자세한 내용은 Amazon Textract 개발자 안내서의 Detecting and Analyzing Text in Multi-Page Documents를 참조하세요.

- 자세한 API 내용은 명령 참조GetDocumentTextDetection의 섹션을 참조하세요. AWS CLI

### `start-document-analysis`

다음 코드 예시에서는 `start-document-analysis`을 사용하는 방법을 보여 줍니다.

AWS CLI

여러 페이지 문서의 텍스트 분석을 시작하는 방법

다음 `start-document-analysis` 예시에서는 여러 페이지가 있는 문서에서 비동기식 텍스트 분석을 시작하는 방법을 보여줍니다.

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"
```

출력:

```
{
    "JobId": "df7cf32ebbd2a5de113535fcf4d921926a701b09b4e7d089f3aebadb41e0712b"
}
```

자세한 내용은 Amazon Textract 개발자 안내서의 Detecting and Analyzing Text in Multi-Page Documents를 참조하세요.

- 자세한 API 내용은 명령 참조StartDocumentAnalysis의 섹션을 참조하세요. AWS CLI

### `start-document-text-detection`

다음 코드 예시에서는 `start-document-text-detection`을 사용하는 방법을 보여 줍니다.

AWS CLI

여러 페이지 문서의 텍스트 감지를 시작하는 방법

다음 `start-document-text-detection` 예시에서는 여러 페이지가 있는 문서에서 비동기식 텍스트 감지를 시작하는 방법을 보여줍니다.

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"
```

출력:

```
{
    "JobId": "57849a3dc627d4df74123dca269d69f7b89329c870c65bb16c9fd63409d200b9"
}
```

자세한 내용은 Amazon Textract 개발자 안내서의 Detecting and Analyzing Text in Multi-Page Documents를 참조하세요.

- 자세한 API 내용은 명령 참조StartDocumentTextDetection의 섹션을 참조하세요. AWS CLI

## 를 사용한 Amazon Transcribe 예제 AWS CLI

다음 코드 예제에서는 Amazon Transcribe 와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `create-language-model`

다음 코드 예시에서는 `create-language-model`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 훈련 및 튜닝 데이터를 모두 사용하여 사용자 지정 언어 모델을 생성합니다.

다음 `create-language-model` 예제에서는 사용자 지정 언어 모델을 생성합니다. 사용자 지정 언어 모델을 사용하여 법률, 접객 서비스, 금융 및 보험과 같은 도메인의 트랜스크립션 성능을 개선

할 수 있습니다. 언어 코드에 유효한 언어 코드를 입력합니다. 의 경우 base-model-name사용자 지정 언어 모델로 트랜스크립션하려는 오디오의 샘플 속도에 가장 적합한 기본 모델을 지정합니다. 모델 이름에 사용자 지정 언어 모델을 호출할 이름을 지정합니다.

```
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"
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 사용자 지정 언어 모델을 사용한 도메인별 트랜스크립션 정확도 개선을 참조하세요.

예제 2: 훈련 데이터만 사용하여 사용자 지정 언어 모델을 생성합니다.

다음 `create-language-model` 예시에서는 오디오 파일을 트랜스크립션합니다. 사용자 지정 언어 모델을 사용하여 법률, 접객 서비스, 금융 및 보험과 같은 도메인의 트랜스크립션 성능을 개선할 수 있습니다. 언어 코드에 유효한 언어 코드를 입력합니다. 의 경우 base-model-name사용자 지정 언어 모델로 트랜스크립션하려는 오디오의 샘플 속도에 가장 적합한 기본 모델을 지정합니다. 모델 이름에 사용자 지정 언어 모델을 호출할 이름을 지정합니다.

```
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"
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 사용자 지정 언어 모델을 사용한 도메인별 트랜스크립션 정확도 개선을 참조하세요.

- 자세한 API 내용은 명령 참조CreateLanguageModel의 섹션을 참조하세요. AWS CLI

### `create-medical-vocabulary`

다음 코드 예시에서는 `create-medical-vocabulary`을 사용하는 방법을 보여 줍니다.

AWS CLI

의료 사용자 지정 어휘를 생성하려면

다음 `create-medical-vocabulary` 예시에서는 사용자 지정 어휘를 생성합니다. 사용자 지정 어휘를 생성하려면 더 정확하게 트랜스크립션하려는 모든 용어가 포함된 텍스트 파일을 만들어야 합니다. 에 해당 텍스트 파일의 Amazon Simple Storage Service(Amazon S3)URI를 vocabulary-file-uri지정합니다. language-code의 경우 사용자 지정 어휘의 언어에 해당하는 언어 코드를 지정합니다. vocabulary-name의 경우 사용자 지정 어휘의 이름을 지정합니다.

```
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
```

출력:

```
{
    "VocabularyName": "cli-medical-vocab-example",
    "LanguageCode": "language-code",
    "VocabularyState": "PENDING"
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 의료 사용자 지정 어휘를 참조하세요.

- 자세한 API 내용은 명령 참조CreateMedicalVocabulary의 섹션을 참조하세요. AWS CLI

### `create-vocabulary-filter`

다음 코드 예시에서는 `create-vocabulary-filter`을 사용하는 방법을 보여 줍니다.

AWS CLI

어휘 필터를 생성하려면

다음 `create-vocabulary-filter` 예제에서는 트랜스크립션에 표시하지 않을 단어 목록이 포함된 텍스트 파일을 사용하는 어휘 필터를 생성합니다. 언어 코드의 경우 어휘 필터의 언어에 해당하는 언어 코드를 지정합니다. 에 텍스트 파일의 Amazon Simple Storage Service(Amazon S3)URI를 vocabulary-filter-file-uri지정합니다. 에서 어휘 필터의 이름을 vocabulary-filter-name지정합니다.

```
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
```

출력:

```
{
    "VocabularyFilterName": "cli-vocabulary-filter-example",
    "LanguageCode": "language-code"
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 원치 않는 단어 필터링을 참조하세요.

- 자세한 API 내용은 명령 참조CreateVocabularyFilter의 섹션을 참조하세요. AWS CLI

### `create-vocabulary`

다음 코드 예시에서는 `create-vocabulary`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 어휘를 생성하는 방법

다음 `create-vocabulary` 예시에서는 사용자 지정 어휘를 생성합니다. 사용자 지정 어휘를 생성하려면 더 정확하게 트랜스크립션하려는 모든 용어가 포함된 텍스트 파일을 만들어야 합니다. 에 해당 텍스트 파일의 Amazon Simple Storage Service(Amazon S3)URI를 vocabulary-file-uri지정합니다. language-code의 경우 사용자 지정 어휘의 언어에 해당하는 언어 코드를 지정합니다. vocabulary-name의 경우 사용자 지정 어휘의 이름을 지정합니다.

```
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
```

출력:

```
{
    "VocabularyName": "cli-vocab-example",
    "LanguageCode": "language-code",
    "VocabularyState": "PENDING"
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 사용자 지정 어휘를 참조하세요.

- 자세한 API 내용은 명령 참조CreateVocabulary의 섹션을 참조하세요. AWS CLI

### `delete-language-model`

다음 코드 예시에서는 `delete-language-model`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 언어 모델을 삭제하려면

다음 delete-language-model 예제에서는 사용자 지정 언어 모델을 삭제합니다.

```
aws transcribe delete-language-model \
    --model-name model-name
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Transcribe 개발자 안내서의 사용자 지정 언어 모델을 사용한 도메인별 트랜스크립션 정확도 개선을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteLanguageModel의 섹션을 참조하세요. AWS CLI

### delete-medical-transcription-job

다음 코드 예시에서는 delete-medical-transcription-job을 사용하는 방법을 보여 줍니다.

AWS CLI

의료 트랜스크립션 작업을 삭제하는 방법

다음 delete-medical-transcription-job 예시에서는 의료 트랜스크립션 작업을 삭제합니다.

```
aws transcribe delete-medical-transcription-job \
    --medical-transcription-job-name medical-transcription-job-name
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Transcribe 개발자 안내서DeleteMedicalTranscriptionJob의 섹션을 참조하세요. Amazon Transcribe

- 자세한 API 내용은 명령 참조DeleteMedicalTranscriptionJob의 섹션을 참조하세요. AWS CLI

### delete-medical-vocabulary

다음 코드 예시에서는 delete-medical-vocabulary을 사용하는 방법을 보여 줍니다.

AWS CLI

의료 사용자 지정 어휘를 삭제하려면

다음 delete-medical-vocabulary 예제에서는 의료 사용자 지정 어휘를 삭제합니다. 어휘 이름에 의료 사용자 지정 어휘의 이름을 지정합니다.

```
aws transcribe delete-vocabulary \
    --vocabulary-name medical-custom-vocabulary-name
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Transcribe 개발자 안내서의 의료 사용자 지정 어휘를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteMedicalVocabulary의 섹션을 참조하세요. AWS CLI

### delete-transcription-job

다음 코드 예시에서는 delete-transcription-job을 사용하는 방법을 보여 줍니다.

AWS CLI

트랜스크립션 작업 중 하나를 삭제하는 방법

다음 delete-transcription-job 예시에서는 트랜스크립션 작업 중 하나를 삭제합니다.

```
aws transcribe delete-transcription-job \
    --transcription-job-name your-transcription-job
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Transcribe 개발자 안내서DeleteTranscriptionJob의 섹션을 참조하세요. Amazon Transcribe

- 자세한 API 내용은 명령 참조DeleteTranscriptionJob의 섹션을 참조하세요. AWS CLI

### delete-vocabulary-filter

다음 코드 예시에서는 delete-vocabulary-filter을 사용하는 방법을 보여 줍니다.

AWS CLI

어휘 필터를 삭제하려면

다음 delete-vocabulary-filter 예제에서는 어휘 필터를 삭제합니다.

```
aws transcribe delete-vocabulary-filter \
    --vocabulary-filter-name vocabulary-filter-name
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Transcribe 개발자 안내서의 원치 않는 단어 필터링을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteVocabularyFilter의 섹션을 참조하세요. AWS CLI

### `delete-vocabulary`

다음 코드 예시에서는 `delete-vocabulary`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 어휘를 삭제하는 방법

다음 `delete-vocabulary` 예시에서는 사용자 지정 어휘를 삭제합니다.

```
aws transcribe delete-vocabulary \
    --vocabulary-name vocabulary-name
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon Transcribe 개발자 안내서의 사용자 지정 어휘를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteVocabulary의 섹션을 참조하세요. AWS CLI

### `describe-language-model`

다음 코드 예시에서는 `describe-language-model`을 사용하는 방법을 보여 줍니다.

AWS CLI

특정 사용자 지정 언어 모델에 대한 정보를 가져오려면

다음 `describe-language-model` 예제에서는 특정 사용자 지정 언어 모델에 대한 정보를 가져옵니다. 예를 들어 아래에서 모델이 NarrowBand 또는 모델을 사용하여 훈련되었는지 여부를 확인할 BaseModelName 수 있습니다 WideBand . NarrowBand 기본 모델이 있는 사용자 지정 언어 모델은 샘플 속도가 16 미만인 오디오를 트랜스크립션할 수 있습니다kHz. WideBand 기본 모델을 사용하는 언어 모델은 샘플 속도가 16보다 큰 오디오를 트랜스크립션할 수 있습니다kHz. S3Uri 파라미터는 사용자 지정 언어 모델을 생성하기 위해 훈련 데이터에 액세스하는 데 사용한 Amazon S3 접두사를 나타냅니다.

```
aws transcribe describe-language-model \
```

```
    --model-name cli-clm-example
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 사용자 지정 언어 모델을 사용한 도메인별 트랜스크립션 정확도 개선을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeLanguageModel의 섹션을 참조하세요. AWS CLI

### get-medical-transcription-job

다음 코드 예시에서는 get-medical-transcription-job을 사용하는 방법을 보여 줍니다.

AWS CLI

특정 의료 트랜스크립션 작업에 대한 정보를 얻으려면

다음 get-medical-transcription-job 예제에서는 특정 의료 트랜스크립션 작업에 대한 정보를 가져옵니다. 트랜스크립션 결과에 액세스하려면 TranscriptFileUri 파라미터를 사용합니다. 트랜스크립션 작업에 추가 기능을 활성화한 경우 설정 객체에서 해당 기능을 볼 수 있습니다. 특수 파라미터는 공급자의 의료 전문 분야를 보여줍니다. 유형 파라미터는 트랜스크립션 작업의 스피치가 의학적 대화인지 또는 의학적 지시인지를 나타냅니다.

```
aws transcribe get-medical-transcription-job \
```

```
    --medical-transcription-job-name vocabulary-dictation-medical-transcription-job
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 배치 트랜스크립션을 참조하세요.

- 자세한 API 내용은 명령 참조GetMedicalTranscriptionJob의 섹션을 참조하세요. AWS CLI

### `get-medical-vocabulary`

다음 코드 예시에서는 `get-medical-vocabulary`을 사용하는 방법을 보여 줍니다.

AWS CLI

의료 사용자 지정 어휘에 대한 정보를 얻으려면

다음 get-medical-vocabulary 예제에서는 의료 사용자 지정 어휘에 대한 정보를 가져옵니다. VocabularyState 파라미터를 사용하여 어휘의 처리 상태를 볼 수 있습니다. 인 경우 StartMedicalTranscriptionJob 작업에 사용할 READY수 있습니다.

```
aws transcribe get-medical-vocabulary \
    --vocabulary-name medical-vocab-example
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 의료 사용자 지정 어휘를 참조하세요.

- 자세한 API 내용은 명령 참조GetMedicalVocabulary의 섹션을 참조하세요. AWS CLI

### get-transcription-job

다음 코드 예시에서는 get-transcription-job을 사용하는 방법을 보여 줍니다.

AWS CLI

특정 트랜스크립션 작업에 대한 정보를 가져오려면

다음 get-transcription-job 예시에서는 특정 트랜스크립션 작업에 대한 정보를 가져옵니다. 트랜스크립션 결과에 액세스하려면 TranscriptFileUri 파라미터를 사용합니다. MediaFileUri 파라미터를 사용하여 이 작업으로 복사한 오디오 파일을 확인합니다. Settings 객체를 사용하여 트랜스크립션 작업에서 활성화한 선택적 기능을 확인할 수 있습니다.

```
aws transcribe get-transcription-job \
    --transcription-job-name your-transcription-job
```

출력:

```
{
```

```
    "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
    }
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 시작하기(AWS 명령줄 인터페이스)를 참조하세요.

- 자세한 API 내용은 명령 참조GetTranscriptionJob의 섹션을 참조하세요. AWS CLI

### `get-vocabulary-filter`

다음 코드 예시에서는 `get-vocabulary-filter`을 사용하는 방법을 보여 줍니다.

AWS CLI

어휘 필터에 대한 정보를 가져오려면

다음 `get-vocabulary-filter` 예제에서는 어휘 필터에 대한 정보를 가져옵니다. DownloadUri 파라미터를 사용하여 어휘 필터를 생성하는 데 사용한 단어 목록을 가져올 수 있습니다.

```
aws transcribe get-vocabulary-filter \
```

```
    --vocabulary-filter-name testFilter
```

출력:

```
{
    "VocabularyFilterName": "testFilter",
    "LanguageCode": "language-code",
    "LastModifiedTime": "2020-05-07T22:39:32.147000+00:00",
    "DownloadUri": "https://Amazon-S3-location-to-download-your-vocabulary-filter"
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 원치 않는 단어 필터링을 참조하세요.

- 자세한 API 내용은 명령 참조GetVocabularyFilter의 섹션을 참조하세요. AWS CLI

### get-vocabulary

다음 코드 예시에서는 get-vocabulary을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 어휘에 대한 정보를 가져오려면

다음 get-vocabulary 예시에서는 이전에 생성한 사용자 지정 어휘에 대한 정보를 가져옵니다.

```
aws transcribe get-vocabulary \
    --vocabulary-name cli-vocab-1
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 사용자 지정 어휘를 참조하세요.

- 자세한 API 내용은 명령 참조GetVocabulary의 섹션을 참조하세요. AWS CLI

### `list-language-models`

다음 코드 예시에서는 `list-language-models`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 언어 모델을 나열하려면

다음 `list-language-models` 예제에서는 AWS 계정 및 리전과 연결된 사용자 지정 언어 모델을 나열합니다. `S3Uri` 및 `TuningDataS3Uri` 파라미터를 사용하여 훈련 데이터 또는 튜닝 데이터로 사용한 Amazon S3 접두사를 찾을 수 있습니다. 는 NarrowBand, 또는 WideBand 모델을 사용하여 사용자 지정 언어 모델을 생성했는지 여부를 BaseModelName 알려줍니다. NarrowBand 기본 모델을 사용하여 사용자 지정 언어 모델을 사용하여 샘플 속도가 16 미만인 오디오 kHz 를 트랜스크립션할 수 있습니다. WideBand 기본 모델을 사용하여 사용자 지정 언어 모델로 오디오 16 kHz 이상을 트랜스크립션할 수 있습니다. `ModelStatus` 파라미터는 트랜스크립션 작업에서 사용자 지정 언어 모델을 사용할 수 있는지 여부를 보여줍니다. 값이 인 경우 COMPLETED트랜스크립션 작업에 사용할 수 있습니다.

```
aws transcribe list-language-models
```

출력:

```
{
    "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"
            }
        }
    ]
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 사용자 지정 언어 모델을 사용한 도메인별 트랜스크립션 정확도 개선을 참조하세요.

- 자세한 API 내용은 명령 참조ListLanguageModels의 섹션을 참조하세요. AWS CLI

### `list-medical-transcription-jobs`

다음 코드 예시에서는 `list-medical-transcription-jobs`을 사용하는 방법을 보여 줍니다.

AWS CLI

의료 트랜스크립션 작업을 나열하는 방법

다음 list-medical-transcription-jobs 예제에서는 AWS 계정 및 리전과 연결된 의료 트랜스크립션 작업을 나열합니다. 특정 트랜스크립션 작업에 대한 자세한 내용을 보려면 트랜스크립션 출력에서 MedicalTranscriptionJobName 파라미터 값을 복사하고 get-medical-transcription-job 명령 MedicalTranscriptionJobName 옵션에 해당 값을 지정합니다. 더 많은 트랜스크립션 작업을 보려면 NextToken 파라미터 값을 복사하고 명령을 list-medical-transcription-jobs 다시 실행한 다음 --next-token 옵션에 해당 값을 지정합니다.

```
aws transcribe list-medical-transcription-jobs
```

출력:

```
{
    "NextToken": "3/PblzkiGhzjER3KHuQt2fmbPLF7cDYafjFMEoGn44ON/
gsuUSTIkGyanvRE6WMXFd/ZTEc2EZj+P9eii/
z1O2FDYli6RLI0WoRX4RwMisVrh9G0Kie0Y8ikBCdtqlZB10Wa9McC+ebOl
+LaDtZPC4u6ttoHLRlEfzqstHXSgapXg3tEBtm9piIaPB6MOM5BB6t86+qtmocTR/
qrteHZBBudhTfbCwhsxaqujHiiUvFdm3BQbKKWIW06yV9b+4f38oD2lVIan
+vfUs3gBYAl5VTDmXXzQPBQOHPjtwmFI+IWX15nSUjWuN3TUylHgPWzDaYT8qBtu0Z+3UG4V6b
+K2CC0XszXg5rBq9hYgNzy4XoFh/6s5DoSnzq49Q9xHgHdT2yBADFmvFK7myZBsj75+2vQZOSVpWUPy3WT/32zFAcoEL
+mFYfUjtTZ8n/jq7aQEjQ42A
+X/7K6JgOcdVPtEg8PlDr5kgYYG3q3OmYXX37U3FZuJmnTI63VtIXsNnOU5eGoYObtpk00Nq9UkzgSJxqj84ZD5n
+S0EGy9ZUYBJRRcGeYUM3Q4DbSJfUwSAqcFdLIWZdp8qIREMQIBWy7BLwSdyqsQo2vRrd53hm5aWM7SVf6pPq6X/
IXR5+1eUOOD8/coaTT4ES2DerbV6RkV4o0VT1d0SdVX/
MmtkNG8nYj8PqU07w7988quh1ZP6D80veJS1q73tUUR9MjnGernW2tAnvnLNhdefBcD
+sZVfYq3iBMFY7wTy1P1G6NqW9GrYDYoX3tTPWlD7phpbVSyKrh/
PdYrps5UxnsGoA1b7L/FfAXDfUoGrGUB4N3JsPYXX9D++g+6gV1qBBs/
WfF934aKqfD6UTggm/zV3GAOWiBpfvAZRvEb924i6yGHyMC7y54O1ZAwSBupmI
+FFd13CaPO4kN1vJlth6aM5vUPXg4BpyUhtbRhwD/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"
        }
    ]
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 https://docs.aws.amazon.com/transcribe/latest/dg/batch-med-transcription.html>을 참조하세요.

- 자세한 API 내용은 명령 참조ListMedicalTranscriptionJobs의 섹션을 참조하세요. AWS CLI

### `list-medical-vocabularies`

다음 코드 예시에서는 `list-medical-vocabularies`을 사용하는 방법을 보여 줍니다.

AWS CLI

의료 사용자 지정 어휘를 나열하려면

다음 `list-medical-vocabularies` 예제에서는 AWS 계정 및 리전과 연결된 의료 사용자 지정 어휘를 나열합니다. 특정 트랜스크립션 작업에 대한 자세한 내용을 보려면 트랜스크립션 출력에 `MedicalTranscriptionJobName` 파라미터 값을 복사하고 `get-medical-transcription-job` 명령 `MedicalTranscriptionJobName` 옵션에 해당 값을 지정합니다. 더 많은 트랜스크립션 작업을 보려면 `NextToken` 파라미터 값을 복사하고 명령을 `list-medical-transcription-jobs` 다시 실행한 다음 `--next-token` 옵션에 해당 값을 지정합니다.

```
aws transcribe list-medical-vocabularies
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 의료 사용자 지정 어휘를 참조하세요.

- 자세한 API 내용은 명령 참조ListMedicalVocabularies의 섹션을 참조하세요. AWS CLI

### `list-transcription-jobs`

다음 코드 예시에서는 `list-transcription-jobs`을 사용하는 방법을 보여 줍니다.

AWS CLI

트랜스크립션 작업을 나열하는 방법

다음 `list-transcription-jobs` 예제에서는 AWS 계정 및 리전과 연결된 트랜스크립션 작업을 나열합니다.

```
aws transcribe list-transcription-jobs
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 시작하기(AWS 명령줄 인터페이스)를 참조하세요.

- 자세한 API 내용은 명령 참조ListTranscriptionJobs의 섹션을 참조하세요. AWS CLI

### `list-vocabularies`

다음 코드 예시에서는 `list-vocabularies`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 어휘를 나열하는 방법

다음 `list-vocabularies` 예제에서는 AWS 계정 및 리전과 연결된 사용자 지정 어휘를 나열합니다.

```
aws transcribe list-vocabularies
```

출력:

```
{
    "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"
```

```
        }
    ]
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 사용자 지정 어휘를 참조하세요.

- 자세한 API 내용은 명령 참조ListVocabularies의 섹션을 참조하세요. AWS CLI

### `list-vocabulary-filters`

다음 코드 예시에서는 `list-vocabulary-filters`을 사용하는 방법을 보여 줍니다.

AWS CLI

어휘 필터를 나열하려면

다음 `list-vocabulary-filters` 예제에서는 AWS 계정 및 리전과 연결된 어휘 필터를 나열합니다.

```
aws transcribe list-vocabulary-filters
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 원치 않는 단어 필터링을 참조하세요.

- 자세한 API 내용은 명령 참조ListVocabularyFilters의 섹션을 참조하세요. AWS CLI

### `start-medical-transcription-job`

다음 코드 예시에서는 `start-medical-transcription-job`을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: 오디오 파일로 저장된 의료 구술을 트랜스크립션하는 방법

다음 `start-medical-transcription-job` 예시에서는 오디오 파일을 트랜스크립션합니다. `OutputBucketName` 파라미터에는 트랜스크립션 출력의 위치를 지정합니다.

```
aws transcribe start-medical-transcription-job \
    --cli-input-json file://myfile.json
```

`myfile.json`의 콘텐츠:

```
{
    "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"
    }
}
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 배치 트랜스크립션 개요를 참조하세요.

예 2: 오디오 파일로 저장된 의사와 환자 간 대화를 트랜스크립션하는 방법

다음 `start-medical-transcription-job` 예시에서는 의사와 환자 간 대화가 포함된 오디오 파일을 트랜스크립션합니다. OutputBucketName 파라미터에서 트랜스크립션 출력의 위치를 지정합니다.

```
aws transcribe start-medical-transcription-job \
    --cli-input-json file://mysecondfile.json
```

`mysecondfile.json`의 콘텐츠:

```
{
    "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"
    }
}
```

출력:

```
{
```

```
    "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"
    }
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 배치 트랜스크립션 개요를 참조하세요.

예 3: 의사와 환자 간 대화의 멀티채널 오디오 파일을 트랜스크립션하는 방법

다음 `start-medical-transcription-job` 예시에서는 오디오 파일에 있는 각 채널의 오디오를 트랜스크립션하고 각 채널의 개별 트랜스크립션을 단일 트랜스크립션 출력으로 병합합니다. `OutputBucketName` 파라미터에는 트랜스크립션 출력의 위치를 지정합니다.

```
aws transcribe start-medical-transcription-job \
    --cli-input-json file://mythirdfile.json
```

`mythirdfile.json`의 콘텐츠:

```
{
    "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
        }
}
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 채널 식별을 참조하세요.

예 4: 의사와 환자 간 대화의 오디오 파일을 트랜스크립션하고 트랜스크립션 출력에서 화자를 식별하는 방법

다음 `start-medical-transcription-job` 예시에서는 오디오 파일을 트랜스크립션하고 트랜스크립션 출력에서 각 화자의 음성에 레이블을 지정합니다. `OutputBucketName` 파라미터에는 트랜스크립션 출력의 위치를 지정합니다.

```
aws transcribe start-medical-transcription-job \
    --cli-input-json file://myfourthfile.json
```

`myfourthfile.json`의 콘텐츠:

```
{
    "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
        }
}
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 화자 식별을 참조하세요.

예 5: 최대 두 개의 대체 트랜스크립션을 사용하여 오디오 파일로 저장된 의료 대화를 트랜스크립션하는 방법

다음 `start-medical-transcription-job` 예시에서는 단일 오디오 파일에서 최대 두 개의 대체 트랜스크립션을 생성합니다. 모든 트랜스크립션에는 신뢰도가 있습니다. 기본적으로 Amazon Transcribe은 신뢰도가 가장 높은 트랜스크립션을 반환합니다. Amazon Transcribe에서 신뢰도가 낮은 추가 트랜스크립션을 반환하도록 지정할 수 있습니다. `OutputBucketName` 파라미터에는 트랜스크립션 출력의 위치를 지정합니다.

```
aws transcribe start-medical-transcription-job \
    --cli-input-json file://myfifthfile.json
```

myfifthfile.json의 콘텐츠:

```
{
    "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
    }
}
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 대체 트랜스크립션을 참조하세요.

예 6: 최대 두 개의 대체 트랜스크립션을 사용하여 의료 구술 오디오 파일을 트랜스크립션하는 방법

다음 `start-medical-transcription-job` 예시에서는 오디오 파일을 트랜스크립션하고 어휘 필터를 사용하여 원하지 않는 단어를 마스킹합니다. OutputBucketName 파라미터에서 트랜스크립션 출력의 위치를 지정합니다.

```
aws transcribe start-medical-transcription-job \
    --cli-input-json file://mysixthfile.json
```

mysixthfile.json의 콘텐츠:

```
{
    "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
    }
}
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 대체 트랜스크립션을 참조하세요.

예 7: 사용자 지정 어휘로 정확도를 높여 의료 구술 오디오 파일을 트랜스크립션하는 방법

다음 `start-medical-transcription-job` 예시에서는 오디오 파일을 트랜스크립션하고 이전에 생성한 의료 사용자 지정 어휘를 사용하여 트랜스크립션 정확도를 높입니다. `OutputBucketName` 파라미터에는 트랜스크립션 출력의 위치를 지정합니다.

```
aws transcribe start-transcription-job \
    --cli-input-json file://myseventhfile.json
```

`mysixthfile.json`의 콘텐츠:

```
{
    "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"
    }
}
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 의료 사용자 지정 어휘를 참조하세요.

- 자세한 API 내용은 명령 참조StartMedicalTranscriptionJob의 섹션을 참조하세요. AWS CLI

### start-transcription-job

다음 코드 예시에서는 start-transcription-job을 사용하는 방법을 보여 줍니다.

AWS CLI

예 1: 오디오 파일을 트랜스크립션하는 방법

다음 start-transcription-job 예시에서는 오디오 파일을 트랜스크립션합니다.

```
aws transcribe start-transcription-job \
    --cli-input-json file://myfile.json
```

myfile.json의 콘텐츠:

```
{
    "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"
    }
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 시작하기(AWS 명령줄 인터페이스)를 참조하세요.

예 2: 다중 채널 오디오 파일을 트랜스크립션하는 방법

다음 `start-transcription-job` 예시에서는 다중 채널 오디오 파일을 트랜스크립션합니다.

```
aws transcribe start-transcription-job \
    --cli-input-json file://mysecondfile.json
```

`mysecondfile.json`의 콘텐츠:

```
{
    "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
    }
}
```

출력:

```
{
    "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
        }
    }
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 다중 채널 오디오 트랜스크립션을 참조하세요.

예 3: 오디오 파일을 트랜스크립션하고 다른 화자를 식별하는 방법

다음 `start-transcription-job` 예시에서는 오디오 파일을 트랜스크립션하고 트랜스크립션 출력에서 화자를 식별합니다.

```
aws transcribe start-transcription-job \
    --cli-input-json file://mythirdfile.json
```

mythirdfile.json의 콘텐츠:

```
{
    "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
    }
}
```

출력:

```
{
    "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
        }
    }
```

```
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 화자 식별을 참조하세요.

예 4: 오디오 파일을 트랜스크립션하고 트랜스크립션 출력에서 원하지 않는 단어를 마스킹하는 방법

다음 `start-transcription-job` 예시에서는 오디오 파일을 트랜스크립션하고 이전에 생성한 어휘 필터를 사용하여 원하지 않는 단어를 마스킹합니다.

```
aws transcribe start-transcription-job \
    --cli-input-json file://myfourthfile.json
```

myfourthfile.json의 콘텐츠:

```
{
    "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"
    }
}
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 트랜스크립션 필터링을 참조하세요.

예 5: 오디오 파일을 트랜스크립션하고 트랜스크립션 출력에서 원하지 않는 단어를 제거하는 방법

다음 `start-transcription-job` 예시에서는 오디오 파일을 트랜스크립션하고 이전에 생성한 어휘 필터를 사용하여 원하지 않는 단어를 마스킹합니다.

```
aws transcribe start-transcription-job \
    --cli-input-json file://myfifthfile.json
```

myfifthfile.json의 콘텐츠:

```
{
    "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"
    }
}
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 트랜스크립션 필터링을 참조하세요.

예 6: 사용자 지정 어휘로 정확도를 높여 오디오 파일을 트랜스크립션하는 방법

다음 `start-transcription-job` 예시에서는 오디오 파일을 트랜스크립션하고 이전에 생성한 어휘 필터를 사용하여 원하지 않는 단어를 마스킹합니다.

```
aws transcribe start-transcription-job \
    --cli-input-json file://mysixthfile.json
```

mysixthfile.json의 콘텐츠:

```
{
    "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"
    }
}
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 트랜스크립션 필터링을 참조하세요.

예 7: 오디오 파일의 언어를 식별하고 트랜스크립션하는 방법

다음 `start-transcription-job` 예시에서는 오디오 파일을 트랜스크립션하고 이전에 생성한 어휘 필터를 사용하여 원하지 않는 단어를 마스킹합니다.

```
aws transcribe start-transcription-job \
    --cli-input-json file://myseventhfile.json
```

myseventhfile.json의 콘텐츠:

```
{
    "TranscriptionJobName": "cli-identify-language-transcription-job",
    "IdentifyLanguage": true,
    "Media": {
        "MediaFileUri": "s3://DOC-EXAMPLE-BUCKET/Amazon-S3-prefix/your-media-file-
name.file-extension"
    }
}
```

출력:

```
{
    "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
    }
```

```
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 언어 식별을 참조하세요.

예 8: 개인 식별 정보를 수정하여 오디오 파일을 트랜스크립션하는 방법

다음 `start-transcription-job` 예시에서는 오디오 파일을 트랜스크립션하고 트랜스크립션 출력에서 개인 식별 정보를 수정합니다.

```
aws transcribe start-transcription-job \
    --cli-input-json file://myeighthfile.json
```

myeigthfile.json의 콘텐츠:

```
{
    "TranscriptionJobName": "cli-redaction-job",
    "LanguageCode": "language-code",
    "Media": {
        "MediaFileUri": "s3://Amazon-S3-Prefix/your-media-file.file-extension"
    },
    "ContentRedaction": {
        "RedactionOutput":"redacted",
        "RedactionType":"PII"
    }
}
```

출력:

```
{
    "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"
        }
    }
```

```
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 자동 콘텐츠 편집을 참조하세요.

예제 9: 개인 식별 정보(PII)가 편집되고 편집되지 않은 트랜스크립트를 생성하는 방법

다음 `start-transcription-job` 예시에서는 오디오 파일의 트랜스크립션 두 개를 생성합니다. 하나는 개인 식별 정보를 수정한 것이고 다른 하나는 수정하지 않은 것입니다.

```
aws transcribe start-transcription-job \
    --cli-input-json file://myninthfile.json
```

myninthfile.json의 콘텐츠:

```
{
    "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"
    }
}
```

출력:

```
{
    "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"
        }
    }
```

```
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 자동 콘텐츠 편집을 참조하세요.

예 10: 이전에 생성한 사용자 지정 언어 모델을 사용하여 오디오 파일을 트랜스크립션하는 방법

다음 `start-transcription-job` 예시에서는 이전에 생성한 사용자 지정 언어 모델을 사용하여 오디오 파일을 트랜스크립션합니다.

```
aws transcribe start-transcription-job \
    --cli-input-json file://mytenthfile.json
```

`mytenthfile.json`의 콘텐츠:

```
{
    "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"
    }
}
```

출력:

```
{
    "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"
        }
    }
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 사용자 지정 언어 모델을 사용한 도메인별 트랜스크립션 정확도 개선을 참조하세요.

- 자세한 API 내용은 명령 참조StartTranscriptionJob의 섹션을 참조하세요. AWS CLI

### update-medical-vocabulary

다음 코드 예시에서는 update-medical-vocabulary을 사용하는 방법을 보여 줍니다.

AWS CLI

의료 사용자 지정 어휘를 새 용어로 업데이트하는 방법.

다음 update-medical-vocabulary 예제는 의료 사용자 지정 어휘에 사용되는 용어를 새 어휘로 대체합니다. 사전 조건: 의료 사용자 지정 어휘의 용어를 바꾸려면 파일을 새 용어로 바꿔야 합니다.

```
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
```

출력:

```
{
    "VocabularyName": "medical-custom-vocabulary",
    "LanguageCode": "en-US",
    "VocabularyState": "PENDING"
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 의료 사용자 지정 어휘를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateMedicalVocabulary의 섹션을 참조하세요. AWS CLI

### update-vocabulary-filter

다음 코드 예시에서는 update-vocabulary-filter을 사용하는 방법을 보여 줍니다.

AWS CLI

어휘 필터의 단어를 바꾸려면

다음 update-vocabulary-filter 예제에서는 어휘 필터의 단어를 새 단어로 바꿉니다. 사전 조건: 어휘 필터를 새 단어로 업데이트하려면 해당 단어를 텍스트 파일로 저장해야 합니다.

```
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
```

출력:

```
{
    "VocabularyFilterName": "vocabulary-filter-name",
    "LanguageCode": "language-code",
    "LastModifiedTime": "2020-09-23T18:40:35.139000+00:00"
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 원치 않는 단어 필터링을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateVocabularyFilter의 섹션을 참조하세요. AWS CLI

### update-vocabulary

다음 코드 예시에서는 update-vocabulary을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 어휘를 새 용어로 업데이트하는 방법

다음 update-vocabulary 예시에서는 사용자 지정 어휘를 생성하는 데 사용된 용어를 사용자가 제공한 새 용어로 덮어씁니다. 사전 조건: 사용자 지정 어휘의 용어를 바꾸려면 새 용어가 포함된 파일이 필요합니다.

```
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
```

출력:

```
{
    "VocabularyName": "custom-vocabulary",
```

```
    "LanguageCode": "language",
    "VocabularyState": "PENDING"
}
```

자세한 내용은 Amazon Transcribe 개발자 안내서의 사용자 지정 어휘를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateVocabulary의 섹션을 참조하세요. AWS CLI

## 를 사용한 Amazon Translate 예제 AWS CLI

다음 코드 예제에서는 Amazon Translate 와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `import-terminology`

다음 코드 예시에서는 `import-terminology`을 사용하는 방법을 보여 줍니다.

AWS CLI

파일에서 사용자 지정 용어를 가져오려면

다음 `import-terminology` 예제에서는 `test-terminology.csv` 파일`MyTestTerminology`에서 라는 용어를 생성합니다.

```
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
```

test-terminology.csv의 콘텐츠:

en,fr,es,zh Hello world!,Bonjour tout le monde!,Hola Mundo!,???
Amazon,Amazon,Amazon,Amazon

출력:

```
{
    "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"
    }
}
```

- 자세한 API 내용은 명령 참조ImportTerminology의 섹션을 참조하세요. AWS CLI

## Trusted Advisor 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 Trusted Advisor.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `get-organization-recommendation`

다음 코드 예시에서는 `get-organization-recommendation`을 사용하는 방법을 보여 줍니다.

AWS CLI

조직 권장 사항을 가져오려면

다음 `get-organization-recommendation` 예제에서는 식별자로 조직 권장 사항을 가져옵니다.

```
aws trustedadvisor get-organization-recommendation \
    --organization-recommendation-identifier arn:aws:trustedadvisor:::organization-recommendation/9534ec9b-bf3a-44e8-8213-2ed68b39d9d5
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 Trusted Advisor API AWS 사용 설명서의 Trusted Advisor 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조GetOrganizationRecommendation의 섹션을 참조하세요. AWS CLI

### `get-recommendation`

다음 코드 예시에서는 `get-recommendation`을 사용하는 방법을 보여 줍니다.

AWS CLI

추천을 받으려면

다음 `get-recommendation` 예제에서는 식별자로 추천을 가져옵니다.

```
aws trustedadvisor get-recommendation \
    --recommendation-
identifier arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-
bbb7-491a-833b-5773e9589578
```

출력:

```
{
    "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/7DAFEmoDos",
        "id": "55fa4d2e-bbb7-491a-833b-5773e9589578",
        "lastUpdatedAt": "2023-11-01T15:57:58.673Z",
        "pillarSpecificAggregates": {
            "costOptimizing": {
                "estimatedMonthlySavings": 0.0,
                "estimatedPercentMonthlySavings": 0.0
```

```
            }
        },
        "pillars": [
            "security"
        ],
        "resourcesAggregates": {
            "errorCount": 1,
            "okCount": 0,
            "warningCount": 0
        },
        "source": "ta_check",
        "status": "error",
        "type": "standard"
    }
}
```

자세한 내용은 Trusted Advisor API AWS 사용 설명서의 Trusted Advisor 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조GetRecommendation의 섹션을 참조하세요. AWS CLI

### `list-checks`

다음 코드 예시에서는 `list-checks`을 사용하는 방법을 보여 줍니다.

AWS CLI

Trusted Advisor 검사를 나열하려면

다음 `list-checks` 예제에서는 Trusted Advisor 검사를 모두 나열합니다.

```
aws trustedadvisor list-checks
```

출력:

```
{
    "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"
}
```

자세한 내용은 Trusted Advisor 사용 설명서API AWS 의 Trusted Advisor 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조ListChecks의 섹션을 참조하세요. AWS CLI

### `list-organization-recommendation-accounts`

다음 코드 예시에서는 `list-organization-recommendation-accounts`을 사용하는 방법을 보여 줍니다.

AWS CLI

조직 추천 계정을 나열하려면

다음 `list-organization-recommendation-accounts` 예제에서는 조직 권장 사항에 대한 모든 계정 권장 사항 요약을 식별자별로 나열합니다.

```
aws trustedadvisor list-organization-recommendation-accounts \
    --organization-recommendation-identifier arn:aws:trustedadvisor:::organization-recommendation/9534ec9b-bf3a-44e8-8213-2ed68b39d9d5
```

출력:

```
{
    "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"
}
```

자세한 내용은 Trusted Advisor 사용 설명서API AWS 의 Trusted Advisor 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조ListOrganizationRecommendationAccounts의 섹션을 참조하세요. AWS CLI

### `list-organization-recommendation-resources`

다음 코드 예시에서는 `list-organization-recommendation-resources`을 사용하는 방법을 보여 줍니다.

AWS CLI

조직 권장 리소스를 나열하려면

다음 `list-organization-recommendation-resources` 예제에서는 조직 권장 사항에 대한 모든 리소스를 식별자별로 나열합니다.

```
aws trustedadvisor list-organization-recommendation-resources \
    --organization-recommendation-identifier arn:aws:trustedadvisor:::organization-recommendation/5a694939-2e54-45a2-ae72-730598fa89d0
```

출력:

```
{
    "organizationRecommendationResourceSummaries": [
        {
            "arn": "arn:aws:trustedadvisor::000000000000:recommendation-resource/5a694939-2e54-45a2-ae72-730598fa89d0/bb38affc0ce0681d9a6cd13f30238ba03a8f63dfe7a379dc403c619119d86af",
            "awsResourceId": "database-1-instance-1",
            "id": "bb38affc0ce0681d9a6cd13f302383ba03a8f63dfe7a379dc403c619119d86af",
            "lastUpdatedAt": "2023-11-01T15:09:51.891Z",
            "metadata": {
                "0": "14",
                "1": "208.79999999999998",
                "2": "database-1-instance-1",
                "3": "db.r5.large",
```

```
                "4": "false",
                "5": "us-west-2",
                "6": "arn:aws:rds:us-west-2:000000000000:db:database-1-instance-1",
                "7": "1"
            },
            "recommendationArn": "arn:aws:trustedadvisor:::organization-
recommendation/5a694939-2e54-45a2-ae72-730598fa89d0",
            "regionCode": "us-west-2",
            "status": "warning"
        },
        {
            "arn": "arn:aws:trustedadvisor::000000000000:recommendation-
resource/5a694939-2e54-45a2-
ae72-730598fa89d0/51fded4d7a3278818df9cfe344ff5762cec46c095a6763d1ba1ba53bd0e1b0e6",
            "awsResourceId": "database-1",
            "id":
 "51fded4d7a3278818df9cfe344ff5762cec46c095a6763d1ba1ba53bd0e1b0e6",
            "lastUpdatedAt": "2023-11-01T15:09:51.891Z",
            "metadata": {
                "0": "14",
                "1": "31.679999999999996",
                "2": "database-1",
                "3": "db.t3.small",
                "4": "false",
                "5": "us-west-2",
                "6": "arn:aws:rds:us-west-2:000000000000:db:database-1",
                "7": "20"
            },
            "recommendationArn": "arn:aws:trustedadvisor:::organization-
recommendation/5a694939-2e54-45a2-ae72-730598fa89d0",
            "regionCode": "us-west-2",
            "status": "warning"
        },
        {
            "arn": "arn:aws:trustedadvisor::000000000000:recommendation-
resource/5a694939-2e54-45a2-ae72-730598fa89d0/
f4d01bd20f4cd5372062aafc8786c489e48f0ead7cdab121463bf9f89e40a36b",
            "awsResourceId": "database-2-instance-1-us-west-2a",
            "id":
 "f4d01bd20f4cd5372062aafc8786c489e48f0ead7cdab121463bf9f89e40a36b",
            "lastUpdatedAt": "2023-11-01T15:09:51.891Z",
            "metadata": {
                "0": "14",
                "1": "187.20000000000002",
```

```
                "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"
}
```

자세한 내용은 Trusted Advisor 사용 설명서API AWS 의 Trusted Advisor 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조ListOrganizationRecommendationResources의 섹션을 참조하세요. AWS CLI

### `list-organization-recommendations`

다음 코드 예시에서는 `list-organization-recommendations`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 조직 권장 사항 나열

다음 `list-organization-recommendations` 예제에서는 모든 조직 권장 사항을 나열하며 필터는 포함하지 않습니다.

```
aws trustedadvisor list-organization-recommendations
```

출력:

```
{
    "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"
}
```

자세한 내용은 Trusted Advisor 사용 설명서APIAWS 의 Trusted Advisor 시작하기를 참조하세요.

예제 2: 필터를 사용하여 조직 권장 사항을 나열하려면

다음 `list-organization-recommendations` 예제는 '보안' 필라의 일부인 최대 하나의 조직 권장 사항을 필터링하고 반환합니다.

```
aws trustedadvisor list-organization-recommendations \
    --pillar security \
    --max-items 100
```

출력:

```
{
    "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"
}
```

자세한 내용은 Trusted Advisor 사용 설명서APIAWS 의 Trusted Advisor 시작하기를 참조하세요.

예제 3: 페이지 매김 토큰을 사용하여 조직 권장 사항을 나열하려면

다음 `list-organization-recommendations` 예제에서는 이전 요청에서 반환된 “nextToken”를 사용하여 조직 권장 사항의 다음 페이지를 가져옵니다.

```
aws trustedadvisor list-organization-recommendations \
    --pillar security \
    --max-items 100 \
    --starting-token <next-token>
```

출력:

```
{
    "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"
    }]
}
```

자세한 내용은 Trusted Advisor API AWS 사용 설명서의 Trusted Advisor 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조ListOrganizationRecommendations의 섹션을 참조하세요. AWS CLI

### `list-recommendation-resources`

다음 코드 예시에서는 `list-recommendation-resources`을 사용하는 방법을 보여 줍니다.

AWS CLI

추천 리소스를 나열하려면

다음 `list-recommendation-resources` 예제에서는 권장 사항에 대한 모든 리소스를 식별자별로 나열합니다.

```
aws trustedadvisor list-recommendation-resources \
    --recommendation-
identifier arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-
bbb7-491a-833b-5773e9589578
```

출력:

```
{
    "recommendationResourceSummaries": [
        {
            "arn": "arn:aws:trustedadvisor::000000000000:recommendation-
resource/55fa4d2e-
bbb7-491a-833b-5773e9589578/18959a1f1973cff8e706e9d9bde28bba36cd602a6b2cb86c8b61252835236010
            "id":
 "18959a1f1973cff8e706e9d9bde28bba36cd602a6b2cb86c8b61252835236010",
            "awsResourceId": "webcms-dev-01",
            "lastUpdatedAt": "2023-11-01T15:09:51.891Z",
            "metadata": {
                "0": "14",
                "1": "123.12000000000002",
                "2": "webcms-dev-01",
                "3": "db.m6i.large",
                "4": "false",
                "5": "us-east-1",
                "6": "arn:aws:rds:us-east-1:000000000000:db:webcms-dev-01",
                "7": "20"
            },
            "recommendationArn":
 "arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-
bbb7-491a-833b-5773e9589578",
            "regionCode": "us-east-1",
            "status": "warning"
        },
        {
```

```
            "arn": "arn:aws:trustedadvisor::000000000000:recommendation-
resource/55fa4d2e-bbb7-491a-833b-5773e9589578/
e6367ff500ac90db8e4adeb4892e39ee9c36bbf812dcbce4b9e4fefcec9eb63e",
            "id":
 "e6367ff500ac90db8e4adeb4892e39ee9c36bbf812dcbce4b9e4fefcec9eb63e",
            "awsResourceId": "aws-dev-db-stack-instance-1",
            "lastUpdatedAt": "2023-11-01T15:09:51.891Z",
            "metadata": {
                "0": "14",
                "1": "29.52",
                "2": "aws-dev-db-stack-instance-1",
                "3": "db.t2.small",
                "4": "false",
                "5": "us-east-1",
                "6": "arn:aws:rds:us-east-1:000000000000:db:aws-dev-db-stack-
instance-1",
                "7": "1"
            },
            "recommendationArn":
 "arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-
bbb7-491a-833b-5773e9589578",
            "regionCode": "us-east-1",
            "status": "warning"
        },
        {
            "arn": "arn:aws:trustedadvisor::000000000000:recommendation-
resource/55fa4d2e-
bbb7-491a-833b-5773e9589578/31aa78ba050a5015d2d38cca7f5f1ce88f70857c4e1c3ad03f8f9fd95dad7459
            "id":
 "31aa78ba050a5015d2d38cca7f5f1ce88f70857c4e1c3ad03f8f9fd95dad7459",
            "awsResourceId": "aws-awesome-apps-stack-db",
            "lastUpdatedAt": "2023-11-01T15:09:51.891Z",
            "metadata": {
                "0": "14",
                "1": "114.48000000000002",
                "2": "aws-awesome-apps-stack-db",
                "3": "db.m6g.large",
                "4": "false",
                "5": "us-east-1",
                "6": "arn:aws:rds:us-east-1:000000000000:db:aws-awesome-apps-stack-
db",
                "7": "100"
            },
```

```
            "recommendationArn":
 "arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-
bbb7-491a-833b-5773e9589578",
            "regionCode": "us-east-1",
            "status": "warning"
        }
    ],
    "nextToken": "REDACTED"
}
```

자세한 내용은 Trusted Advisor 사용 설명서APIAWS 의 Trusted Advisor 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조ListRecommendationResources의 섹션을 참조하세요. AWS CLI

### `list-recommendations`

다음 코드 예시에서는 `list-recommendations`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 권장 사항 나열

다음 `list-recommendations` 예제에서는 모든 권장 사항을 나열하고 필터는 포함하지 않습니다.

```
aws trustedadvisor list-recommendations
```

출력:

```
{
    "recommendationSummaries": [
        {
            "arn": "arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-
bbb7-491a-833b-5773e9589578",
            "name": "MFA Recommendation",
            "awsServices": [
                "iam"
            ],
            "checkArn": "arn:aws:trustedadvisor:::check/7DAFEmoDos",
            "id": "55fa4d2e-bbb7-491a-833b-5773e9589578",
            "lastUpdatedAt": "2023-11-01T15:57:58.673Z",
            "pillarSpecificAggregates": {
                "costOptimizing": {
```

```
                    "estimatedMonthlySavings": 0.0,
                    "estimatedPercentMonthlySavings": 0.0
                }
            },
            "pillars": [
                "security"
            ],
            "resourcesAggregates": {
                "errorCount": 1,
                "okCount": 0,
                "warningCount": 0
            },
            "source": "ta_check",
            "status": "error",
            "type": "standard"
        },
        {
            "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"
}
```

자세한 내용은 Trusted Advisor 사용 설명서API AWS 의 Trusted Advisor 시작하기를 참조하세요.

예제 2: 필터를 사용하여 추천을 나열하려면

다음 `list-recommendations` 예제에서는 권장 사항을 나열하고 필터를 포함합니다.

```
aws trustedadvisor list-recommendations \
    --aws-service iam \
    --max-items 100
```

출력:

```
{
    "recommendationSummaries": [{
        "arn": "arn:aws:trustedadvisor::000000000000:recommendation/55fa4d2e-
bbb7-491a-833b-5773e9589578",
        "name": "MFA Recommendation",
        "awsServices": [
            "iam"
        ],
        "checkArn": "arn:aws:trustedadvisor:::check/7DAFEmoDos",
        "id": "55fa4d2e-bbb7-491a-833b-5773e9589578",
        "lastUpdatedAt": "2023-11-01T15:57:58.673Z",
        "pillarSpecificAggregates": {
            "costOptimizing": {
                "estimatedMonthlySavings": 0.0,
                "estimatedPercentMonthlySavings": 0.0
            }
        },
        "pillars": [
            "security"
        ],
        "resourcesAggregates": {
            "errorCount": 1,
            "okCount": 0,
            "warningCount": 0
        },
        "source": "ta_check",
```

```
        "status": "error",
        "type": "standard"
    }],
    "nextToken": "REDACTED"
}
```

자세한 내용은 Trusted Advisor 사용 설명서API AWS 의 Trusted Advisor 시작하기를 참조하세요.

예제 3: 페이지 매김 토큰을 사용하여 권장 사항을 나열하려면

다음 `list-recommendations` 예제에서는 이전 요청에서 반환된 “nextToken”를 사용하여 필터링된 권장 사항의 다음 페이지를 가져옵니다.

```
aws trustedadvisor list-recommendations \
    --aws-service rds \
    --max-items 100 \
    --starting-token <next-token>
```

출력:

```
{
    "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"
    }]
}
```

자세한 내용은 Trusted Advisor 사용 설명서API AWS 의 Trusted Advisor 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조ListRecommendations의 섹션을 참조하세요. AWS CLI

### update-organization-recommendation-lifecycle

다음 코드 예시에서는 update-organization-recommendation-lifecycle을 사용하는 방법을 보여 줍니다.

AWS CLI

조직 권장 사항 수명 주기를 업데이트하려면

다음 update-organization-recommendation-lifecycle 예제에서는 조직 권장 사항의 수명 주기를 식별자별로 업데이트합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Trusted Advisor API AWS 사용 설명서의 Trusted Advisor 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateOrganizationRecommendationLifecycle의 섹션을 참조하세요. AWS CLI

### update-recommendation-lifecycle

다음 코드 예시에서는 update-recommendation-lifecycle을 사용하는 방법을 보여 줍니다.

AWS CLI

권장 사항 수명 주기를 업데이트하려면

다음 `update-recommendation-lifecycle` 예제에서는 권장 사항의 수명 주기를 식별자별로 업데이트합니다.

```
aws trustedadvisor update-recommendation-lifecycle \
    --recommendation-
identifier arn:aws:trustedadvisor::000000000000:recommendation/861c9c6e-
f169-405a-8b59-537a8caccd7a \
    --lifecycle-stage resolved \
    --update-reason-code valid_business_case
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Trusted Advisor 사용 설명서API AWS 의 Trusted Advisor 시작하기를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateRecommendationLifecycle의 섹션을 참조하세요. AWS CLI

## 를 사용하여 확인된 권한 예제 AWS CLI

다음 코드 예제에서는 Verified Permissions와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `create-identity-source`

다음 코드 예시에서는 `create-identity-source`을 사용하는 방법을 보여 줍니다.

AWS CLI

자격 증명 소스를 생성하려면

다음 `create-identity-source` 예제에서는 지정된 Amazon Cognito 사용자 풀에 저장된 자격 증명을 참조할 수 있는 자격 증명 소스를 생성합니다. 이러한 자격 증명은 Verified Permissions에서 유형의 엔터티로 사용할 수 있습니다`User`.

```
aws verifiedpermissions create-identity-source \
    --configuration file://config.txt \
    --principal-entity-type "User" \
    --policy-store-id PSEXAMPLEabcdefg111111
```

`config.txt`의 콘텐츠:

```
{
    "cognitoUserPoolConfiguration": {
        "userPoolArn": "arn:aws:cognito-idp:us-west-2:123456789012:userpool/us-west-2_1a2b3c4d5",
        "clientIds":["a1b2c3d4e5f6g7h8i9j0kalbmc"]
    }
}
```

출력:

```
{
    "createdDate": "2023-05-19T20:30:28.214829+00:00",
    "identitySourceId": "ISEXAMPLEabcdefg111111",
    "lastUpdatedDate": "2023-05-19T20:30:28.214829+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
}
```

자격 증명 소스에 대한 자세한 내용은 Amazon Verified Permissions 사용 설명서의 자격 증명 공급자와 함께 Amazon Verified Permissions 사용을 참조하세요.

- 자세한 API 내용은 명령 참조CreateIdentitySource의 섹션을 참조하세요. AWS CLI

### `create-policy-store`

다음 코드 예시에서는 `create-policy-store`을 사용하는 방법을 보여 줍니다.

AWS CLI

정책 스토어를 생성하려면

다음 create-policy-store 예제에서는 현재 AWS 리전에 정책 스토어를 생성합니다.

```
aws verifiedpermissions create-policy-store \
    --validation-settings "mode=STRICT"
```

출력:

```
{
    "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"
}
```

정책 스토어에 대한 자세한 내용은 Amazon Verified Permissions 사용 설명서의 Amazon Verified Permissions 정책 스토어를 참조하세요.

- 자세한 API 내용은 명령 참조CreatePolicyStore의 섹션을 참조하세요. AWS CLI

### create-policy-template

다음 코드 예시에서는 create-policy-template을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 정책 템플릿 생성

다음 create-policy-template 예제에서는 보안 주체의 자리 표시자가 포함된 문이 포함된 정책 템플릿을 생성합니다.

```
aws verifiedpermissions create-policy-template \
    --definition file://template1.txt \
    --policy-store-id PSEXAMPLEabcdefg111111
```

template1.txt 파일의 콘텐츠:

```
permit(
```

```
    principal in ?principal,
    action == Action::"view",
    resource == Photo::"VacationPhoto94.jpg"
);
```

출력:

```
{
    "createdDate": "2023-06-12T20:47:42.804511+00:00",
    "lastUpdatedDate": "2023-06-12T20:47:42.804511+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyTemplateId": "PTEXAMPLEabcdefg111111"
}
```

정책 템플릿에 대한 자세한 내용은 Amazon Verified Permissions 사용 설명서의 Amazon Verified Permissions 정책 템플릿을 참조하세요.

- 자세한 API 내용은 명령 참조CreatePolicyTemplate의 섹션을 참조하세요. AWS CLI

### `create-policy`

다음 코드 예시에서는 `create-policy`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 정적 정책 생성

다음 `create-policy` 예제에서는 보안 주체와 리소스를 모두 지정하는 정책 범위를 사용하여 정적 정책을 생성합니다.

```
aws verifiedpermissions create-policy \
    --definition file://definition1.txt \
    --policy-store-id PSEXAMPLEabcdefg111111
```

`definition1.txt` 파일의 콘텐츠:

```
{
    "static": {
        "description":  "Grant everyone of janeFriends UserGroup access to the
 vacationFolder Album",
        "statement": "permit(principal in UserGroup::\"janeFriends\", action,
 resource in Album::\"vacationFolder\" );"
```

```
    }
}
```

출력:

```
{
    "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"
    }
}
```

예제 2: 모든 사용자에게 리소스에 대한 액세스 권한을 부여하는 정적 정책을 생성하려면

다음 `create-policy` 예제에서는 리소스만 지정하는 정책 범위를 사용하여 정적 정책을 생성합니다.

```
aws verifiedpermissions create-policy \
    --definition file://definition2.txt \
    --policy-store-id PSEXAMPLEabcdefg111111
```

`definition2.txt` 파일의 콘텐츠:

```
{
    "static": {
        "description":  "Grant everyone access to the publicFolder Album",
        "statement": "permit(principal, action, resource in Album::\"publicFolder
\");"
    }
}
```

출력:

```
{
    "createdDate": "2023-06-12T20:39:44.975897+00:00",
    "lastUpdatedDate": "2023-06-12T20:39:44.975897+00:00",
    "policyId": "PbfR73F8oh5MMfr9uRtFDB",
    "policyStoreId": "PSEXAMPLEabcdefg222222",
    "policyType": "STATIC",
    "resource": {
        "entityId": "publicFolder",
        "entityType": "Album"
    }
}
```

예제 3: 지정된 템플릿과 연결된 템플릿 연결 정책을 생성하려면

다음 `create-policy` 예제에서는 지정된 정책 템플릿을 사용하여 템플릿 연결 정책을 생성하고 지정된 보안 주체를 새 템플릿 연결 정책과 함께 사용하도록 연결합니다.

```
aws verifiedpermissions create-policy \
    --definition file://definition.txt \
    --policy-store-id PSEXAMPLEabcdefg111111
```

`definition.txt`의 콘텐츠:

```
{
    "templateLinked": {
        "policyTemplateId": "PTEXAMPLEabcdefg111111",
        "principal": {
            "entityType": "User",
            "entityId": "alice"
        }
    }
}
```

출력:

```
{
    "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"
    }
}
```

정책에 대한 자세한 내용은 Amazon Verified Permissions 사용 설명서의 Amazon Verified Permissions 정책을 참조하세요.

- 자세한 API 내용은 명령 참조CreatePolicy의 섹션을 참조하세요. AWS CLI

### `delete-identity-source`

다음 코드 예시에서는 `delete-identity-source`을 사용하는 방법을 보여 줍니다.

AWS CLI

자격 증명 소스를 삭제하려면

다음 `delete-identity-source` 예제에서는 지정된 ID가 있는 자격 증명 소스를 삭제합니다.

```
aws verifiedpermissions delete-identity-source \
    --identity-source-id ISEXAMPLEabcdefg111111 \
    --policy-store-id PSEXAMPLEabcdefg111111
```

이 명령은 출력을 생성하지 않습니다.

자격 증명 소스에 대한 자세한 내용은 Amazon Verified Permissions 사용 설명서의 자격 증명 공급자와 함께 Amazon Verified Permissions 사용을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteIdentitySource의 섹션을 참조하세요. AWS CLI

### `delete-policy-store`

다음 코드 예시에서는 `delete-policy-store`을 사용하는 방법을 보여 줍니다.

AWS CLI

정책 스토어를 삭제하려면

다음 delete-policy-store 예제에서는 지정된 ID가 있는 정책 스토어를 삭제합니다.

```
aws verifiedpermissions delete-policy-store \
    --policy-store-id PSEXAMPLEabcdefg111111
```

이 명령은 출력을 생성하지 않습니다.

정책 스토어에 대한 자세한 내용은 Amazon Verified Permissions 사용 설명서의 Amazon Verified Permissions 정책 스토어를 참조하세요.

- 자세한 API 내용은 명령 참조DeletePolicyStore의 섹션을 참조하세요. AWS CLI

### delete-policy-template

다음 코드 예시에서는 delete-policy-template을 사용하는 방법을 보여 줍니다.

AWS CLI

정책 템플릿을 삭제하려면

다음 delete-policy-template 예제에서는 지정된 ID가 있는 정책 템플릿을 삭제합니다.

```
aws verifiedpermissions delete-policy \
    --policy-template-id PTEXAMPLEabcdefg111111 \
    --policy-store-id PSEXAMPLEabcdefg111111
```

이 명령은 출력을 생성하지 않습니다.

정책 템플릿에 대한 자세한 내용은 Amazon Verified Permissions 사용 설명서의 Amazon Verified Permissions 정책 템플릿을 참조하세요.

- 자세한 API 내용은 명령 참조DeletePolicyTemplate의 섹션을 참조하세요. AWS CLI

### delete-policy

다음 코드 예시에서는 delete-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

정적 또는 템플릿 연결 정책을 삭제하려면

다음 delete-policy 예제에서는 지정된 ID가 있는 정책을 삭제합니다.

```
aws verifiedpermissions delete-policy \
    --policy-id SPEXAMPLEabcdefg111111 \
    --policy-store-id PSEXAMPLEabcdefg111111
```

이 명령은 출력을 생성하지 않습니다.

정책에 대한 자세한 내용은 Amazon Verified Permissions 사용 설명서의 Amazon Verified Permissions 정책을 참조하세요.

- 자세한 API 내용은 명령 참조DeletePolicy의 섹션을 참조하세요. AWS CLI

### get-identity-source

다음 코드 예시에서는 get-identity-source을 사용하는 방법을 보여 줍니다.

AWS CLI

자격 증명 소스에 대한 세부 정보를 검색하려면

다음 get-identity-source 예제에서는 지정된 ID가 있는 자격 증명 소스에 대한 세부 정보를 표시합니다.

```
aws verifiedpermissions get-identity-source \
    --identity-source  ISEXAMPLEabcdefg111111 \
    --policy-store-id PSEXAMPLEabcdefg111111
```

출력:

```
{
    "createdDate": "2023-06-12T22:27:49.150035+00:00",
    "details": {
        "clientIds": [ "a1b2c3d4e5f6g7h8i9j0kalbmc" ],
        "discoveryUrl": "https://cognito-idp.us-west-2.amazonaws.com/us-
west-2_1a2b3c4d5",
        "openIdIssuer": "COGNITO",
        "userPoolArn": "arn:aws:cognito-idp:us-west-2:123456789012:userpool/us-
west-2_1a2b3c4d5"
    },
    "identitySourceId": "ISEXAMPLEabcdefg111111",
    "lastUpdatedDate": "2023-06-12T22:27:49.150035+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
```

```
    "principalEntityType": "User"
}
```

자격 증명 소스에 대한 자세한 내용은 Amazon Verified Permissions 사용 설명서의 자격 증명 공급자와 함께 Amazon Verified Permissions 사용을 참조하세요.

- 자세한 API 내용은 명령 참조GetIdentitySource의 섹션을 참조하세요. AWS CLI

### get-policy-store

다음 코드 예시에서는 get-policy-store을 사용하는 방법을 보여 줍니다.

AWS CLI

정책 스토어에 대한 세부 정보를 검색하려면

다음 get-policy-store 예제에서는 지정된 ID가 있는 정책 스토어에 대한 세부 정보를 표시합니다.

```
aws verifiedpermissions get-policy-store \
    --policy-store-id PSEXAMPLEabcdefg111111
```

출력:

```
{
    "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" }
}
```

정책 스토어에 대한 자세한 내용은 Amazon Verified Permissions 사용 설명서의 Amazon Verified Permissions 정책 스토어를 참조하세요.

- 자세한 API 내용은 명령 참조GetPolicyStore의 섹션을 참조하세요. AWS CLI

### get-policy-template

다음 코드 예시에서는 get-policy-template을 사용하는 방법을 보여 줍니다.

AWS CLI

정책 템플릿에 대한 세부 정보를 검색하려면

다음 `get-policy-template` 예제에서는 지정된 ID를 사용하여 정책 템플릿에 대한 세부 정보를 표시합니다.

```
aws verifiedpermissions get-policy-template \
    --policy-template-id  PTEXAMPLEabcdefg111111 \
    --policy-store-id PSEXAMPLEabcdefg111111
```

출력:

```
{
    "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::
\"view\",\n    resource == Photo::\"VacationPhoto94.jpg\"\n);"
}
```

정책 템플릿에 대한 자세한 내용은 Amazon Verified Permissions 사용 설명서의 Amazon Verified Permissions 정책 템플릿을 참조하세요.

- 자세한 API 내용은 명령 참조GetPolicyTemplate의 섹션을 참조하세요. AWS CLI

### `get-policy`

다음 코드 예시에서는 `get-policy`을 사용하는 방법을 보여 줍니다.

AWS CLI

정책에 대한 세부 정보를 검색하려면

다음 `get-policy` 예제에서는 지정된 ID로 정책의 세부 정보를 표시합니다.

```
aws verifiedpermissions get-policy \
    --policy-id PSEXAMPLEabcdefg111111 \
    --policy-store-id PSEXAMPLEabcdefg111111
```

출력:

```
{
    "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"
    }
}
```

정책에 대한 자세한 내용은 Amazon Verified Permissions 사용 설명서의 Amazon Verified Permissions 정책을 참조하세요.

- 자세한 API 내용은 명령 참조GetPolicy의 섹션을 참조하세요. AWS CLI

### get-schema

다음 코드 예시에서는 get-schema을 사용하는 방법을 보여 줍니다.

AWS CLI

정책 스토어에서 스키마를 검색하려면

다음 get-schema 예제에서는 지정된 정책 스토어의 스키마 세부 정보를 표시합니다.

```
aws verifiedpermissions get-schema \
    --policy-store-id PSEXAMPLEabcdefg111111
```

출력:

```
{
    "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"
}
```

스키마에 대한 자세한 내용은 Amazon Verified Permissions 사용 설명서의 정책 스토어 스키마를 참조하세요.

- 자세한 API 내용은 명령 참조GetSchema의 섹션을 참조하세요. AWS CLI

### `is-authorized-with-token`

다음 코드 예시에서는 `is-authorized-with-token`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 사용자 요청에 대한 권한 부여 결정을 요청하려면(허용)

다음 `is-authorized-with-token` 예제에서는 Amazon Cognito 에서 인증한 사용자에 대한 권한 부여 결정을 요청합니다. 요청은 액세스 토큰이 아닌 Cognito에서 제공하는 자격 증명 토큰을 사용합니다. 이 예제에서는 지정된 정보 스토어가 보안 주체를 유형의 엔터티로 반환하도록 구성됩니다`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"
```

정책 스토어에는 지정된 Cognito 사용자 풀 및 애플리케이션 ID의 ID를 허용하는 다음 문이 있는 정책이 포함되어 있습니다.

```
permit(
    principal == CognitoUser::"us-east-1_1a2b3c4d5|a1b2c3d4e5f6g7h8i9j0kalbmc",
```

```
    action,
    resource == Photo::"VacationPhoto94.jpg"
);
```

출력:

```
{
    "decision":"Allow",
    "determiningPolicies":[
        {
        "determiningPolicyId":"SPEXAMPLEabcdefg111111"
        }
    ],
    "errors":[]
}
```

Cognito 사용자 풀의 자격 증명 사용에 대한 자세한 내용은 Amazon Verified Permissions 사용 설명서의 자격 증명 공급자와 Amazon Verified Permissions 사용을 참조하세요.

- 자세한 API 내용은 명령 참조IsAuthorizedWithToken의 섹션을 참조하세요. AWS CLI

### `is-authorized`

다음 코드 예시에서는 `is-authorized`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 사용자 요청에 대한 권한 부여 결정을 요청하려면(허용)

다음 `is-authorized` 예제에서는 `User`라는 유형의 리소스에 대해 `updatePhoto` 작업을 수행`Alice`하려는 `Photo`라는 유형의 보안 주체에 대한 권한 부여 결정을 요청합니다`VacationPhoto94.jpg`.

응답은 요청이 하나의 정책에서 허용됨을 보여줍니다.

```
aws verifiedpermissions is-authorized \
    --principal entityType=User,entityId=alice \
    --action actionType=Action,actionId=view \
    --resource entityType=Photo,entityId=VactionPhoto94.jpg \
    --policy-store-id PSEXAMPLEabcdefg111111
```

출력:

```
{
    "decision": "ALLOW",
    "determiningPolicies": [
        {
            "policyId": "SPEXAMPLEabcdefg111111"
        }
    ],
    "errors": []
}
```

예제 2: 사용자 요청에 대한 권한 부여 결정을 요청하려면(거부)

다음 예제는 보안 주체가 라는 점을 제외하고 이전 예제와 동일합니다`User::"Bob"`. 정책 스토어에는 해당 사용자가 에 액세스할 수 있도록 허용하는 정책이 포함되어 있지 않습니다`Album::"alice_folder"`.

출력은 목록이 비어 있기 때문에 `DeterminingPolicies`가 암시적`Deny`임을 나타냅니다.

```
aws verifiedpermissions create-policy \
    --definition file://definition2.txt \
    --policy-store-id PSEXAMPLEabcdefg111111
```

출력:

```
{
    "decision": "DENY",
    "determiningPolicies": [],
    "errors": []
}
```

자세한 내용은 Amazon Verified Permissions 사용 설명서 를 참조하세요.

- 자세한 API 내용은 명령 참조IsAuthorized의 섹션을 참조하세요. AWS CLI

### `list-identity-sources`

다음 코드 예시에서는 `list-identity-sources`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 자격 증명 소스를 나열하려면

다음 `list-identity-sources` 예제에서는 지정된 정책 스토어의 모든 자격 증명 소스를 나열합니다.

```
aws verifiedpermissions list-identity-sources \
    --policy-store-id PSEXAMPLEabcdefg111111
```

출력:

```
{
    "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": "ISEXAMPLEabcdefg111111",
            "lastUpdatedDate": "2023-06-12T22:27:49.150035+00:00",
            "policyStoreId": "PSEXAMPLEabcdefg111111",
            "principalEntityType": "User"
        }
    ]
}
```

자격 증명 소스에 대한 자세한 내용은 Amazon Verified Permissions 사용 설명서의 자격 증명 공급자와 함께 Amazon Verified Permissions 사용을 참조하세요.

- 자세한 API 내용은 명령 참조ListIdentitySources의 섹션을 참조하세요. AWS CLI

### `list-policies`

다음 코드 예시에서는 `list-policies`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 정책을 나열하려면

다음 `list-policies` 예제에서는 지정된 정책 스토어의 모든 정책을 나열합니다.

```
aws verifiedpermissions list-policies \
    --policy-store-id PSEXAMPLEabcdefg111111
```

출력:

```
{
    "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"
            }
        }
    ]
}
```

정책에 대한 자세한 내용은 Amazon Verified Permissions 사용 설명서의 Amazon Verified Permissions 정책을 참조하세요.

- 자세한 API 내용은 명령 참조ListPolicies의 섹션을 참조하세요. AWS CLI

### `list-policy-stores`

다음 코드 예시에서는 `list-policy-stores`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 정책 스토어를 나열하려면

다음 `list-policy-stores` 예제에서는 AWS 리전의 모든 정책 스토어를 나열합니다. `create-policy-store` 및 를 제외한 Verified Permissions에 대한 모든 명령은 작업하려는 정책 스토어의 ID를 지정`list-policy-stores`해야 합니다.

```
aws verifiedpermissions list-policy-stores
```

출력:

```
{
    "policyStores": [
        {
            "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg111111",
            "createdDate": "2023-06-05T20:16:46.225598+00:00",
            "policyStoreId": "PSEXAMPLEabcdefg111111"
        },
        {
            "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg222222",
            "createdDate": "2023-06-08T18:09:37.364356+00:00",
            "policyStoreId": "PSEXAMPLEabcdefg222222"
        },
        {
            "arn": "arn:aws:verifiedpermissions::123456789012:policy-store/
PSEXAMPLEabcdefg333333",
            "createdDate": "2023-06-08T18:09:46.920600+00:00",
            "policyStoreId": "PSEXAMPLEabcdefg333333"
        }
    ]
}
```

정책 스토어에 대한 자세한 내용은 Amazon Verified Permissions 사용 설명서의 Amazon Verified Permissions 정책 스토어를 참조하세요.

- 자세한 API 내용은 명령 참조ListPolicyStores의 섹션을 참조하세요. AWS CLI

### `list-policy-templates`

다음 코드 예시에서는 `list-policy-templates`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용 가능한 정책 템플릿을 나열하려면

다음 `list-policy-templates` 예제에서는 지정된 정책 스토어의 모든 정책 템플릿을 나열합니다.

```
aws verifiedpermissions list-policy-templates \
```

```
    --policy-store-id PSEXAMPLEabcdefg111111
```

출력:

```
{
    "policyTemplates": [
        {
            "createdDate": "2023-06-12T20:47:42.804511+00:00",
            "lastUpdatedDate": "2023-06-12T20:47:42.804511+00:00",
            "policyStoreId": "PSEXAMPLEabcdefg111111",
            "policyTemplateId": "PTEXAMPLEabcdefg111111"
        }
    ]
}
```

정책 템플릿에 대한 자세한 내용은 Amazon Verified Permissions 사용 설명서의 Amazon Verified Permissions 정책 템플릿을 참조하세요.

- 자세한 API 내용은 명령 참조ListPolicyTemplates의 섹션을 참조하세요. AWS CLI

**put-schema**

다음 코드 예시에서는 put-schema을 사용하는 방법을 보여 줍니다.

AWS CLI

스키마를 정책 스토어에 저장하려면

다음 put-schema 예제에서는 지정된 정책 스토어에서 스키마를 생성하거나 바꿉니다.

입력 파일의 cedarJson 파라미터는 JSON 객체의 문자열 표현을 취합니다. 여기에는 가장 바깥쪽의 따옴표 쌍 내에 포함된 따옴표(“)가 포함되어 있습니다. 이렇게 하려면 모든 포함된 따옴표 앞에 백슬래시 문자( ')를 붙이고 모든 줄을 줄 바꿈 없이 단일 텍스트 줄로 결합하여 를 JSON 문자열로 변환해야 합니다.

여기서는 가독성을 위해 여러 줄에 걸쳐 예제 문자열을 표시할 수 있지만, 작업을 수행하려면 파라미터를 단일 줄 문자열로 제출해야 합니다.

aws verifiedpermissions put-schema --정의 파일://schema.txt --policy-store-id PSEXAMPLEabcdefg111111

schema.txt의 콘텐츠:

```
{
    "cedarJson": "{\"MySampleNamespace\": {\"actions\": {\"remoteAccess\": {
            \"appliesTo\": {\"principalTypes\": [\"Employee\"]}}},\"entityTypes\": {
            \"Employee\": {\"shape\": {\"attributes\": {\"jobLevel\": {\"type\":
            \"Long\"},\"name\": {\"type\": \"String\"}},\"type\": \"Record\"}}}}}"
}
```

출력:

```
{
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "namespaces": [
        "MySampleNamespace"
    ],
    "createdDate": "2023-06-14T17:47:13.999885+00:00",
    "lastUpdatedDate": "2023-06-14T17:47:13.999885+00:00"
}
```

스키마에 대한 자세한 내용은 Amazon Verified Permissions 사용 설명서의 정책 스토어 스키마를 참조하세요.

- 자세한 API 내용은 명령 참조PutSchema의 섹션을 참조하세요. AWS CLI

### update-identity-source

다음 코드 예시에서는 update-identity-source을 사용하는 방법을 보여 줍니다.

AWS CLI

자격 증명 소스를 업데이트하려면

다음 update-identity-source 예제에서는 새 Cognito 사용자 풀 구성을 제공하고 자격 증명 소스에서 반환한 엔터티 유형을 변경하여 지정된 자격 증명 소스를 수정합니다.

```
aws verifiedpermissions update-identity-source
    --identity-source-id ISEXAMPLEabcdefg111111 \
    --update-configuration file://config.txt \
    --principal-entity-type "Employee" \
    --policy-store-id PSEXAMPLEabcdefg111111
```

config.txt의 콘텐츠:

```
{
        "cognitoUserPoolConfiguration": {
                "userPoolArn": "arn:aws:cognito-idp:us-west-2:123456789012:userpool/
us-west-2_1a2b3c4d5",
                "clientIds":["a1b2c3d4e5f6g7h8i9j0kalbmc"]
        }
}
```

출력:

```
{
    "createdDate": "2023-05-19T20:30:28.214829+00:00",
    "identitySourceId": "ISEXAMPLEabcdefg111111",
    "lastUpdatedDate": "2023-05-19T20:30:28.214829+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111"
}
```

자격 증명 소스에 대한 자세한 내용은 Amazon Verified Permissions 사용 설명서의 자격 증명 공급자와 함께 Amazon Verified Permissions 사용을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateIdentitySource의 섹션을 참조하세요. AWS CLI

### update-policy-store

다음 코드 예시에서는 update-policy-store을 사용하는 방법을 보여 줍니다.

AWS CLI

정책 스토어를 업데이트하려면

다음 update-policy-store 예제에서는 검증 설정을 변경하여 정책 스토어를 수정합니다.

```
aws verifiedpermissions update-policy-store \
    --validation-settings "mode=STRICT" \
    --policy-store-id PSEXAMPLEabcdefg111111
```

출력:

```
{
    "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"
}
```

정책 스토어에 대한 자세한 내용은 Amazon Verified Permissions 사용 설명서의 Amazon Verified Permissions 정책 스토어를 참조하세요.

- 자세한 API 내용은 명령 참조UpdatePolicyStore의 섹션을 참조하세요. AWS CLI

### update-policy-template

다음 코드 예시에서는 update-policy-template을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 정책 템플릿 업데이트

다음 update-policy-template 예제에서는 지정된 템플릿 연결 정책을 수정하여 정책 문을 바꿉니다.

```
aws verifiedpermissions update-policy-template \
    --policy-template-id PTEXAMPLEabcdefg111111 \
    --statement file://template1.txt \
    --policy-store-id PSEXAMPLEabcdefg111111
```

template1.txt 파일의 콘텐츠:

```
permit(
    principal in ?principal,
    action == Action::"view",
    resource == Photo::"VacationPhoto94.jpg"
);
```

출력:

```
{
    "createdDate": "2023-06-12T20:47:42.804511+00:00",
    "lastUpdatedDate": "2023-06-12T20:47:42.804511+00:00",
    "policyStoreId": "PSEXAMPLEabcdefg111111",
    "policyTemplateId": "PTEXAMPLEabcdefg111111"
```

```
}
```

정책 템플릿에 대한 자세한 내용은 Amazon Verified Permissions 사용 설명서의 Amazon Verified Permissions 정책 템플릿을 참조하세요.

- 자세한 API 내용은 명령 참조UpdatePolicyTemplate의 섹션을 참조하세요. AWS CLI

### update-policy

다음 코드 예시에서는 update-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 정적 정책 생성

다음 create-policy 예제에서는 보안 주체와 리소스를 모두 지정하는 정책 범위를 사용하여 정적 정책을 생성합니다.

```
aws verifiedpermissions create-policy \
    --definition file://definition.txt \
    --policy-store-id PSEXAMPLEabcdefg111111
```

statement 파라미터는 JSON 객체의 문자열 표현을 취합니다. 여기에는 가장 바깥쪽의 따옴표 쌍 내에 포함된 따옴표(“)가 포함되어 있습니다. 이렇게 하려면 모든 포함된 따옴표 앞에 백슬래시 문자( ')를 붙이고 모든 줄을 줄 바꿈 없이 단일 텍스트 줄로 결합하여 를 JSON 문자열로 변환해야 합니다.

여기서는 가독성을 위해 여러 줄에 걸쳐 예제 문자열을 표시할 수 있지만, 작업을 수행하려면 파라미터를 단일 줄 문자열로 제출해야 합니다.

definition.txt 파일의 콘텐츠:

```
{
    "static": {
        "description":  "Grant everyone of janeFriends UserGroup access to the
 vacationFolder Album",
        "statement": "permit(principal in UserGroup::\"janeFriends\", action,
 resource in Album::\"vacationFolder\" );"
    }
}
```

출력:

```
{
    "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"
    }
}
```

예제 2: 모든 사용자에게 리소스에 대한 액세스 권한을 부여하는 정적 정책을 생성하려면

다음 `create-policy` 예제에서는 리소스만 지정하는 정책 범위를 사용하여 정적 정책을 생성합니다.

```
aws verifiedpermissions create-policy \
    --definition file://definition2.txt \
    --policy-store-id PSEXAMPLEabcdefg111111
```

`definition2.txt` 파일의 콘텐츠:

```
{
    "static": {
        "description":  "Grant everyone access to the publicFolder Album",
        "statement": "permit(principal, action, resource in Album::\"publicFolder
\");"
    }
}
```

출력:

```
{
    "createdDate": "2023-06-12T20:39:44.975897+00:00",
    "lastUpdatedDate": "2023-06-12T20:39:44.975897+00:00",
    "policyId": "PbfR73F8oh5MMfr9uRtFDB",
```

```
    "policyStoreId": "PSEXAMPLEabcdefg222222",
    "policyType": "STATIC",
    "resource": {
        "entityId": "publicFolder",
        "entityType": "Album"
    }
}
```

예제 3: 지정된 템플릿과 연결된 템플릿 연결 정책을 생성하려면

다음 create-policy 예제에서는 지정된 정책 템플릿을 사용하여 템플릿 연결 정책을 생성하고 지정된 보안 주체를 새 템플릿 연결 정책과 함께 사용하도록 연결합니다.

```
aws verifiedpermissions create-policy \
    --definition file://definition2.txt \
    --policy-store-id PSEXAMPLEabcdefg111111
```

definition3.txt의 내용:

```
{
    "templateLinked": {
        "policyTemplateId": "PTEXAMPLEabcdefg111111",
        "principal": {
            "entityType": "User",
            "entityId": "alice"
        }
    }
}
```

출력:

```
{
    "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"
    }
}
```

정책에 대한 자세한 내용은 Amazon Verified Permissions 사용 설명서의 Amazon Verified Permissions 정책을 참조하세요.

- 자세한 API 내용은 명령 참조UpdatePolicy의 섹션을 참조하세요. AWS CLI

## VPC 를 사용한 Lattice 예제 AWS CLI

다음 코드 예제에서는 VPC Lattice AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `create-listener`

다음 코드 예시에서는 `create-listener`을 사용하는 방법을 보여 줍니다.

AWS CLI

리스너를 생성하려면

다음 `create-listener` 예제에서는 지정된 VPC Lattice 대상 그룹에 트래픽을 전달하는 기본 규칙을 사용하여 HTTPS 리스너를 생성합니다.

```
aws vpc-lattice create-listener \
    --name my-service-listener \
    --protocol HTTPS \
```

```
    --port 443 \
    --service-identifier svc-0285b53b2eEXAMPLE \
    --default-action file://listener-config.json
```

listener-config.json의 콘텐츠:

```
{
    "forward": {
        "targetGroups": [
            {
                "targetGroupIdentifier": "tg-0eaa4b9ab4EXAMPLE"
            }
        ]
    }
}
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon VPC Lattice 사용 설명서의 리스너를 참조하세요.

- 자세한 API 내용은 명령 참조CreateListener의 섹션을 참조하세요. AWS CLI

### create-service-network-service-association

다음 코드 예시에서는 create-service-network-service-association을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스 연결을 생성하려면

다음 create-service-network-service-association 예제에서는 지정된 서비스를 지정된 서비스 네트워크와 연결합니다.

```
aws vpc-lattice create-service-network-service-association \
    --service-identifier svc-0285b53b2eEXAMPLE \
    --service-network-identifier sn-080ec7dc93EXAMPLE
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon VPC Lattice 사용 설명서의 서비스 연결 관리를 참조하세요.

- 자세한 API 내용은 명령 참조CreateServiceNetworkServiceAssociation의 섹션을 참조하세요. AWS CLI

### create-service-network-vpc-association

다음 코드 예시에서는 create-service-network-vpc-association을 사용하는 방법을 보여 줍니다.

AWS CLI

VPC 연결을 생성하려면

다음 create-service-network-vpc-association 예제에서는 지정된 vpc를 지정된 서비스 네트워크와 연결합니다. 지정된 보안 그룹은 에서 서비스 네트워크 및 해당 서비스에 액세스할 VPC 수 있는 리소스를 제어합니다.

```
aws vpc-lattice create-service-network-vpc-association \
    --vpc-identifier vpc-0a1b2c3d4eEXAMPLE \
    --service-network-identifier sn-080ec7dc93EXAMPLE \
    --security-group-ids sg-0aee16bc6cEXAMPLE
```

출력:

```
{
    "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetworkvpcassociation/
snva-0821fc8631EXAMPLE",
    "createdBy": "123456789012",
    "id": "snva-0821fc8631EXAMPLE",
    "securityGroupIds": [
        "sg-0aee16bc6cEXAMPLE"
    ],
    "status": "CREATE_IN_PROGRESS"
}
```

자세한 내용은 Amazon VPC Lattice 사용 설명서의 VPC 연결 관리를 참조하세요.

- 자세한 API 내용은 명령 참조CreateServiceNetworkVpcAssociation의 섹션을 참조하세요. AWS CLI

### create-service-network

다음 코드 예시에서는 create-service-network을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스 네트워크를 생성하려면

다음 create-service-network 예제에서는 지정된 이름으로 서비스 네트워크를 생성합니다.

```
aws vpc-lattice create-service-network \
```

```
    --name my-service-network
```

출력:

```
{
    "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetwork/
sn-080ec7dc93EXAMPLE",
    "authType": "NONE",
    "id": "sn-080ec7dc93EXAMPLE",
    "name": "my-service-network"
}
```

자세한 내용은 Amazon VPC Lattice 사용 설명서의 서비스 네트워크를 참조하세요.

- 자세한 API 내용은 명령 참조CreateServiceNetwork의 섹션을 참조하세요. AWS CLI

### create-service

다음 코드 예시에서는 create-service을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스를 생성하려면

다음 create-service 예제에서는 지정된 이름의 서비스를 생성합니다.

```
aws vpc-lattice create-service \
    --name my-lattice-service
```

출력:

```
{
    "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"
```

```
}
```

자세한 내용은 Amazon VPC Lattice 사용 설명서의 Lattice 서비스를 참조하세요. VPC

- 자세한 API 내용은 명령 참조CreateService의 섹션을 참조하세요. AWS CLI

### `create-target-group`

다음 코드 예시에서는 `create-target-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 유형의 대상 그룹을 생성하려면 INSTANCE

다음 `create-target-group` 예제에서는 지정된 이름, 유형 및 구성을 사용하여 대상 그룹을 생성합니다.

```
aws vpc-lattice create-target-group \
    --name my-lattice-target-group-instance \
    --type INSTANCE \
    --config file://tg-config.json
```

`tg-config.json`의 콘텐츠:

```
{
    "port": 443,
    "protocol": "HTTPS",
    "protocolVersion": "HTTP1",
    "vpcIdentifier": "vpc-f1663d9868EXAMPLE"
}
```

출력:

```
{
    "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"
}
```

예제 2: IP 유형의 대상 그룹을 생성하려면

다음 `create-target-group` 예제에서는 지정된 이름, 유형 및 구성을 사용하여 대상 그룹을 생성합니다.

```
aws vpc-lattice create-target-group \
    --name my-lattice-target-group-ip \
    --type IP \
    --config file://tg-config.json
```

`tg-config.json`의 콘텐츠:

```
{
    "ipAddressType": "IPV4",
    "port": 443,
    "protocol": "HTTPS",
    "protocolVersion": "HTTP1",
    "vpcIdentifier": "vpc-f1663d9868EXAMPLE"
}
```

출력:

```
{
```

```
    "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"
}
```

예제 3: 유형의 대상 그룹을 생성하려면 LAMBDA

다음 `create-target-group` 예제에서는 지정된 이름, 유형 및 구성을 사용하여 대상 그룹을 생성합니다.

```
aws vpc-lattice create-target-group \
    --name my-lattice-target-group-lambda \
    --type LAMBDA
```

출력:

```
{
    "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"
}
```

예제 4: 유형의 대상 그룹을 생성하려면 ALB

다음 `create-target-group` 예제에서는 지정된 이름, 유형 및 구성을 사용하여 대상 그룹을 생성합니다.

```
aws vpc-lattice create-target-group \
    --name my-lattice-target-group-alb \
    --type ALB \
    --config file://tg-config.json
```

`tg-config.json`의 콘텐츠:

```
{
    "port": 443,
    "protocol": "HTTPS",
    "protocolVersion": "HTTP1",
    "vpcIdentifier": "vpc-f1663d9868EXAMPLE"
}
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon VPC Lattice 사용 설명서의 대상 그룹을 참조하세요.

- 자세한 API 내용은 명령 참조CreateTargetGroup의 섹션을 참조하세요. AWS CLI

### delete-auth-policy

다음 코드 예시에서는 delete-auth-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

인증 정책을 삭제하려면

다음 delete-auth-policy 예제에서는 지정된 서비스에 대한 인증 정책을 삭제합니다.

```
aws vpc-lattice delete-auth-policy \
    --resource-identifier svc-0285b53b2eEXAMPLE
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon VPC Lattice 사용 설명서의 인증 정책을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteAuthPolicy의 섹션을 참조하세요. AWS CLI

### delete-listener

다음 코드 예시에서는 delete-listener을 사용하는 방법을 보여 줍니다.

AWS CLI

리스너를 삭제하려면

다음 delete-listener 예제에서는 지정된 리스너를 삭제합니다.

```
aws vpc-lattice delete-listener \
    --listener-identifier listener-07cc7fb0abEXAMPLE \
    --service-identifier svc-0285b53b2eEXAMPLE
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon VPC Lattice 사용 설명서의 리스너를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteListener의 섹션을 참조하세요. AWS CLI

### delete-service-network-service-association

다음 코드 예시에서는 delete-service-network-service-association을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스 연결을 삭제하려면

다음 delete-service-network-service-association 예제에서는 지정된 서비스 연결을 연결 해제합니다.

```
aws vpc-lattice delete-service-network-service-association \
    --service-network-service-association-identifier snsa-031fabb4d8EXAMPLE
```

출력:

```
{
    "arn": "arn:aws:vpc-lattice:us-
east-2:123456789012:servicenetworkserviceassociation/snsa-031fabb4d8EXAMPLE",
    "id": "snsa-031fabb4d8EXAMPLE",
    "status": "DELETE_IN_PROGRESS"
}
```

자세한 내용은 Amazon VPC Lattice 사용 설명서의 서비스 연결 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteServiceNetworkServiceAssociation의 섹션을 참조하세요. AWS CLI

### delete-service-network-vpc-association

다음 코드 예시에서는 delete-service-network-vpc-association을 사용하는 방법을 보여 줍니다.

AWS CLI

VPC 연결을 삭제하려면

다음 delete-service-network-vpc-association 예제에서는 지정된 VPC 연결을 연결 해제합니다.

```
aws vpc-lattice delete-service-network-vpc-association \
```

```
    --service-network-vpc-association-identifier snva-0821fc8631EXAMPLE
```

출력:

```
{
    "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetworkvpcassociation/
snva-0821fc8631EXAMPLE",
    "id": "snva-0821fc8631EXAMPLE",
    "status": "DELETE_IN_PROGRESS"
}
```

자세한 내용은 Amazon VPC Lattice 사용 설명서의 VPC 연결 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteServiceNetworkVpcAssociation의 섹션을 참조하세요. AWS CLI

### delete-service-network

다음 코드 예시에서는 delete-service-network을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스 네트워크를 삭제하려면

다음 delete-service-network 예제에서는 지정된 서비스 네트워크를 삭제합니다.

```
aws vpc-lattice delete-service-network \
    --service-network-identifier sn-080ec7dc93EXAMPLE
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon VPC Lattice 사용 설명서의 서비스 네트워크를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteServiceNetwork의 섹션을 참조하세요. AWS CLI

### delete-service

다음 코드 예시에서는 delete-service을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스를 삭제하는 방법

다음 delete-service 예제에서는 지정된 서비스를 삭제합니다.

```
aws vpc-lattice delete-service \
    --service-identifier svc-0285b53b2eEXAMPLE
```

출력:

```
{
    "arn": "arn:aws:vpc-lattice:us-west-2:123456789012:service/
svc-0285b53b2eEXAMPLE",
    "id": "svc-0285b53b2eEXAMPLE",
    "name": "my-lattice-service",
    "status": "DELETE_IN_PROGRESS"
}
```

자세한 내용은 Amazon VPC Lattice 사용 설명서의 Lattice 서비스를 참조하세요. VPC

- 자세한 API 내용은 명령 참조DeleteService의 섹션을 참조하세요. AWS CLI

### delete-target-group

다음 코드 예시에서는 delete-target-group을 사용하는 방법을 보여 줍니다.

AWS CLI

대상 그룹을 삭제하는 방법

다음 delete-target-group 예시에서는 지정된 대상 그룹을 삭제합니다.

```
aws vpc-lattice delete-target-group \
    --target-group-identifier tg-0eaa4b9ab4EXAMPLE
```

출력:

```
{
    "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:targetgroup/
tg-0eaa4b9ab4EXAMPLE",
    "id": "tg-0eaa4b9ab4EXAMPLE",
    "status": "DELETE_IN_PROGRESS"
}
```

자세한 내용은 Amazon VPC Lattice 사용 설명서의 대상 그룹을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteTargetGroup의 섹션을 참조하세요. AWS CLI

### deregister-targets

다음 코드 예시에서는 deregister-targets을 사용하는 방법을 보여 줍니다.

AWS CLI

대상 등록을 취소하려면

다음 deregister-targets 예제에서는 지정된 대상 그룹에서 지정된 대상을 등록 취소합니다.

```
aws vpc-lattice deregister-targets \
    --targets i-07dd579bc5EXAMPLE \
    --target-group-identifier tg-0eaa4b9ab4EXAMPLE
```

출력:

```
{
    "successful": [
        {
            "id": "i-07dd579bc5EXAMPLE",
            "port": 443
        }
    ],
    "unsuccessful": []
}
```

자세한 내용은 Amazon VPC Lattice 사용 설명서의 대상 등록을 참조하세요.

- 자세한 API 내용은 명령 참조DeregisterTargets의 섹션을 참조하세요. AWS CLI

### get-auth-policy

다음 코드 예시에서는 get-auth-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

인증 정책에 대한 정보를 가져오려면

다음 get-auth-policy 예제에서는 지정된 서비스의 인증 정책에 대한 정보를 가져옵니다.

```
aws vpc-lattice get-auth-policy \
    --resource-identifier svc-0285b53b2eEXAMPLE
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon VPC Lattice 사용 설명서의 인증 정책을 참조하세요.

- 자세한 API 내용은 명령 참조GetAuthPolicy의 섹션을 참조하세요. AWS CLI

### get-listener

다음 코드 예시에서는 get-listener을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스 리스너에 대한 정보를 가져오려면

다음 get-listener 예제에서는 지정된 서비스에 대해 지정된 리스너에 대한 정보를 가져옵니다.

```
aws vpc-lattice get-listener \
    --listener-identifier listener-0ccf55918cEXAMPLE \
    --service-identifier svc-0285b53b2eEXAMPLE
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon VPC Lattice 사용 설명서의 라우팅 정의를 참조하세요.

- 자세한 API 내용은 명령 참조GetListener의 섹션을 참조하세요. AWS CLI

### get-service-network-service-association

다음 코드 예시에서는 get-service-network-service-association을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스 연결에 대한 정보를 가져오려면

다음 get-service-network-service-association 예제에서는 지정된 서비스 연결에 대한 정보를 가져옵니다.

```
aws vpc-lattice get-service-network-service-association \
    --service-network-service-association-identifier snsa-031fabb4d8EXAMPLE
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon VPC Lattice 사용 설명서의 서비스 연결 관리를 참조하세요.

- 자세한 API 내용은 명령 참조GetServiceNetworkServiceAssociation의 섹션을 참조하세요. AWS CLI

### get-service-network-vpc-association

다음 코드 예시에서는 get-service-network-vpc-association을 사용하는 방법을 보여 줍니다.

AWS CLI

VPC 연결에 대한 정보를 가져오려면

다음 get-service-network-vpc-association 예제에서는 지정된 VPC 연결에 대한 정보를 가져옵니다.

```
aws vpc-lattice get-service-network-vpc-association \
    --service-network-vpc-association-identifier snva-0821fc8631EXAMPLE
```

출력:

```
{
```

```
    "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"
}
```

자세한 내용은 Amazon VPC Lattice 사용 설명서의 VPC 연결 관리를 참조하세요.

- 자세한 API 내용은 명령 참조GetServiceNetworkVpcAssociation의 섹션을 참조하세요. AWS CLI

### `get-service-network`

다음 코드 예시에서는 `get-service-network`을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스 네트워크에 대한 정보를 가져오려면

다음 `get-service-network` 예제에서는 지정된 서비스 네트워크에 대한 정보를 가져옵니다.

```
aws vpc-lattice get-service-network \
    --service-network-identifier sn-080ec7dc93EXAMPLE
```

출력:

```
{
   "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
}
```

자세한 내용은 Amazon VPC Lattice 사용 설명서의 서비스 네트워크를 참조하세요.

- 자세한 API 내용은 명령 참조GetServiceNetwork의 섹션을 참조하세요. AWS CLI

### get-service

다음 코드 예시에서는 get-service을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스에 대한 정보를 가져오려면

다음 get-service 예제에서는 지정된 서비스에 대한 정보를 가져옵니다.

```
aws vpc-lattice get-service \
    --service-identifier svc-0285b53b2eEXAMPLE
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon VPC Lattice 사용 설명서의 서비스를 참조하세요.

- 자세한 API 내용은 명령 참조GetService의 섹션을 참조하세요. AWS CLI

### get-target-group

다음 코드 예시에서는 get-target-group을 사용하는 방법을 보여 줍니다.

AWS CLI

대상 그룹에 대한 정보를 가져오려면

다음 get-target-group 예제에서는 대상 유형이 인 지정된 대상 그룹에 대한 정보를 가져옵니다INSTANCE.

```
aws vpc-lattice get-target-group \
    --target-group-identifier tg-0eaa4b9ab4EXAMPLE
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon VPC Lattice 사용 설명서의 대상 그룹을 참조하세요.

- 자세한 API 내용은 명령 참조GetTargetGroup의 섹션을 참조하세요. AWS CLI

### `list-listeners`

다음 코드 예시에서는 `list-listeners`을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스 리스너를 나열하려면

다음 `list-listeners` 예제에서는 지정된 서비스의 리스너를 나열합니다.

```
aws vpc-lattice list-listeners \
    --service-identifier svc-0285b53b2eEXAMPLE
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon VPC Lattice 사용 설명서의 라우팅 정의를 참조하세요.

- 자세한 API 내용은 명령 참조ListListeners의 섹션을 참조하세요. AWS CLI

### `list-service-network-service-associations`

다음 코드 예시에서는 `list-service-network-service-associations`을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스 연결을 나열하려면

다음 `list-service-network-service-associations` 예제에서는 지정된 서비스 네트워크에 대한 서비스 연결을 나열합니다. `--query` 옵션은 출력 범위를 서비스 연결IDs의 로 지정합니다.

```
aws vpc-lattice list-service-network-service-associations \
    --service-network-identifier sn-080ec7dc93EXAMPLE \
    --query items[*].id
```

출력:

```
[
    "snsa-031fabb4d8EXAMPLE",
    "snsa-0e16955a8cEXAMPLE"
]
```

자세한 내용은 Amazon VPC Lattice 사용 설명서의 서비스 연결 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ListServiceNetworkServiceAssociations의 섹션을 참조하세요. AWS CLI

### `list-service-network-vpc-associations`

다음 코드 예시에서는 `list-service-network-vpc-associations`을 사용하는 방법을 보여 줍니다.

AWS CLI

VPC 연결을 나열하려면

다음 `list-service-network-vpc-associations` 예제에서는 지정된 서비스 네트워크의 VPC 연결을 나열합니다. `--query` 옵션은 VPC 연결IDs의 로 출력 범위를 지정합니다.

```
aws vpc-lattice list-service-network-vpc-associations \
    --service-network-identifier sn-080ec7dc93EXAMPLE \
    --query items[*].id
```

출력:

```
[
    "snva-0821fc8631EXAMPLE",
    "snva-0c5dcb60d6EXAMPLE"
]
```

자세한 내용은 Amazon VPC Lattice 사용 설명서의 VPC 연결 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ListServiceNetworkVpcAssociations의 섹션을 참조하세요. AWS CLI

### `list-service-networks`

다음 코드 예시에서는 `list-service-networks`을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스 네트워크를 나열하려면

다음 `list-service-networks` 예제에서는 호출 계정과 소유하거나 공유하는 서비스 네트워크를 나열합니다. 이 `--query` 옵션은 서비스 네트워크의 Amazon 리소스 이름(ARN)으로 결과의 범위를 지정합니다.

```
aws vpc-lattice list-service-networks \
    --query items[*].arn
```

출력:

```
[
    "arn:aws:vpc-lattice:us-east-2:123456789012:servicenetwork/
sn-080ec7dc93EXAMPLE",
    "arn:aws:vpc-lattice:us-east-2:111122223333:servicenetwork/sn-0ec4d436cfEXAMPLE"
```

```
]
```

자세한 내용은 Amazon VPC Lattice 사용 설명서의 서비스 네트워크를 참조하세요.

- 자세한 API 내용은 명령 참조ListServiceNetworks의 섹션을 참조하세요. AWS CLI

### `list-services`

다음 코드 예시에서는 `list-services`을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스를 나열하려면

다음 `list-services` 예제에서는 호출 계정과 소유하거나 공유하는 서비스를 나열합니다. 이 `--query` 옵션은 서비스의 Amazon 리소스 이름(ARN)으로 결과의 범위를 지정합니다.

```
aws vpc-lattice list-services \
    --query items[*].arn
```

출력:

```
[
    "arn:aws:vpc-lattice:us-east-2:123456789012:service/svc-0285b53b2eEXAMPLE",
    "arn:aws:vpc-lattice:us-east-2:111122223333:service/svc-0b8ac96550EXAMPLE"
]
```

자세한 내용은 Amazon VPC Lattice 사용 설명서의 서비스를 참조하세요.

- 자세한 API 내용은 명령 참조ListServices의 섹션을 참조하세요. AWS CLI

### `list-target-groups`

다음 코드 예시에서는 `list-target-groups`을 사용하는 방법을 보여 줍니다.

AWS CLI

대상 그룹을 나열하려면

다음 `list-target-groups` 예제에서는 대상 유형이 인 대상 그룹을 나열합니다LAMBDA.

```
aws vpc-lattice list-target-groups \
```

```
    --target-group-type LAMBDA
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon VPC Lattice 사용 설명서의 대상 그룹을 참조하세요.

- 자세한 API 내용은 명령 참조ListTargetGroups의 섹션을 참조하세요. AWS CLI

### `list-targets`

다음 코드 예시에서는 `list-targets`을 사용하는 방법을 보여 줍니다.

AWS CLI

대상 그룹의 대상을 나열하려면

다음 `list-targets` 예제에서는 지정된 대상 그룹의 대상을 나열합니다.

```
aws vpc-lattice list-targets \
    --target-group-identifier tg-0eaa4b9ab4EXAMPLE
```

출력:

```
{
    "items": [
        {
            "id": "i-07dd579bc5EXAMPLE",
            "port": 443,
            "status": "HEALTHY"
        },
        {
            "id": "i-047b3c9078EXAMPLE",
            "port": 443,
            "reasonCode": "HealthCheckFailed",
            "status": "UNHEALTHY"
        }
    ]
}
```

자세한 내용은 Amazon VPC Lattice 사용 설명서의 대상 그룹을 참조하세요.

- 자세한 API 내용은 명령 참조ListTargets의 섹션을 참조하세요. AWS CLI

### put-auth-policy

다음 코드 예시에서는 put-auth-policy을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스에 대한 인증 정책을 생성하려면

다음 put-auth-policy 예제에서는 지정된 IAM 역할을 사용하는 인증된 보안 주체의 요청에 대한 액세스 권한을 부여합니다. 리소스는 정책ARN이 연결된 서비스의 입니다.

```
aws vpc-lattice put-auth-policy \
    --resource-identifier svc-0285b53b2eEXAMPLE \
    --policy file://auth-policy.json
```

auth-policy.json의 콘텐츠:

```
{
    "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"
        }
    ]
}
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon VPC Lattice 사용 설명서의 인증 정책을 참조하세요.

- 자세한 API 내용은 명령 참조PutAuthPolicy의 섹션을 참조하세요. AWS CLI

### register-targets

다음 코드 예시에서는 register-targets을 사용하는 방법을 보여 줍니다.

AWS CLI

대상을 등록하려면

다음 register-targets 예제에서는 지정된 대상을 지정된 대상 그룹에 등록합니다.

```
aws vpc-lattice register-targets \
    --targets id=i-047b3c9078EXAMPLE id=i-07dd579bc5EXAMPLE \
    --target-group-identifier tg-0eaa4b9ab4EXAMPLE
```

출력:

```
{
```

```
    "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
        }
    ]
}
```

자세한 내용은 Amazon VPC Lattice 사용 설명서의 대상 등록을 참조하세요.

- 자세한 API 내용은 명령 참조RegisterTargets의 섹션을 참조하세요. AWS CLI

## AWS WAF Classic 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS WAF Classic.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `put-logging-configuration`

다음 코드 예시에서는 `put-logging-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 Kinesis Firehose 스트림을 ACL ARN 사용하여 웹에 대한 로깅 구성을 생성하려면 ARN

다음 put-logging-configuration 예제에서는 를 WAF 사용하여 에 대한 로깅 구성을 표시합니다 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=[]
```

출력:

```
{
    "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"
        ]
    }
}
```

- 자세한 API 내용은 명령 참조PutLoggingConfiguration의 섹션을 참조하세요. AWS CLI

### update-byte-match-set

다음 코드 예시에서는 update-byte-match-set을 사용하는 방법을 보여 줍니다.

AWS CLI

바이트 일치 세트를 업데이트하려면

다음 update-byte-match-set 명령은 에서 ByteMatchTuple 객체(필터)를 삭제합니다 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
```

자세한 내용은 AWS WAF 개발자 안내서의 문자열 일치 조건 작업을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateByteMatchSet의 섹션을 참조하세요. AWS CLI

### update-ip-set

다음 코드 예시에서는 update-ip-set을 사용하는 방법을 보여 줍니다.

AWS CLI

IP 세트를 업데이트하려면

다음 update-ip-set 명령은 IPv4 주소로 IPSet를 업데이트하고 IPv6 주소를 삭제합니다.

```
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
```

또는 JSON 파일을 사용하여 입력을 지정할 수 있습니다. 예:

```
aws waf update-ip-set --ip-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90 --change-
token 12cs345-67cd-890b-1cd2-c3a4567d89f1  --updates file://change.json
```

JSON 파일의 콘텐츠가 있는 위치:

```
[
{
"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"
}
}
]
```

자세한 내용은 AWS WAF 개발자 안내서의 IP 매치 조건 작업을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateIpSet의 섹션을 참조하세요. AWS CLI

### update-rule

다음 코드 예시에서는 update-rule을 사용하는 방법을 보여 줍니다.

AWS CLI

규칙을 업데이트하려면

다음 update-rule 명령은 규칙에서 예측 객체를 삭제합니다.

```
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"}
```

자세한 내용은 AWS WAF 개발자 안내서의 규칙 작업을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateRule의 섹션을 참조하세요. AWS CLI

### update-size-constraint-set

다음 코드 예시에서는 update-size-constraint-set을 사용하는 방법을 보여 줍니다.

AWS CLI

크기 제약 조건 세트를 업데이트하려면

다음 update-size-constraint-set 명령은 크기 제약 조건 세트의 SizeConstraint 객체(필터)를 삭제합니다.

```
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
```

자세한 내용은 AWS WAF 개발자 안내서의 크기 제약 조건 작업을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateSizeConstraintSet의 섹션을 참조하세요. AWS CLI

### update-sql-injection-match-set

다음 코드 예시에서는 update-sql-injection-match-set을 사용하는 방법을 보여 줍니다.

AWS CLI

SQL 주입 일치 세트를 업데이트하려면

다음 update-sql-injection-match-set 명령은 SQL 주입 일치 세트의 SqlInjectionMatchTuple 객체(필터)를 삭제합니다.

```
aws waf update-sql-injection-match-set --sql-injection-
match-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90 --
change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 --updates
 Action="DELETE",SqlInjectionMatchTuple={FieldToMatch={Type="QUERY_STRING"},TextTransformati
```

자세한 내용은 AWS WAF 개발자 안내서의 SQL 주사 일치 조건 작업을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateSqlInjectionMatchSet의 섹션을 참조하세요. AWS CLI

### update-web-acl

다음 코드 예시에서는 update-web-acl을 사용하는 방법을 보여 줍니다.

AWS CLI

웹을 업데이트하려면 ACL

다음 update-web-acl 명령은 웹 에서 ActivatedRule 객체를 삭제합니다ACL.

aws waf 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=REGULAR'}"

출력:

```
{
    "ChangeToken": "12cs345-67cd-890b-1cd2-c3a4567d89f1"
}
```

자세한 내용은 , AWS WAF AWS 방화벽 관리자 및 AWS Shield 고급 개발자 안내서의 웹 작업을 ACLs 참조하세요.

- 자세한 API 내용은 명령 참조UpdateWebAcl의 섹션을 참조하세요. AWS CLI

**update-xss-match-set**

다음 코드 예시에서는 update-xss-match-set을 사용하는 방법을 보여 줍니다.

AWS CLI

를 업데이트하려면 XSSMatchSet

다음 update-xss-match-set 명령은 에서 XssMatchTuple 객체(필터)를 삭제합니다. 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
```

자세한 내용은 AWS WAF 개발자 안내서의 교차 사이트 스크립팅 일치 조건 작업을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateXssMatchSet의 섹션을 참조하세요. AWS CLI

## AWS WAF Classic Regional 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS WAF Classic Regional.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `associate-web-acl`

다음 코드 예시에서는 `associate-web-acl`을 사용하는 방법을 보여 줍니다.

AWS CLI

웹을 리소스ACL에 연결하려면

다음 `associate-web-acl` 명령은 에서 ACL지정한 웹 을 resource-arn에서 지정한 리소스 web-acl-id와 연결합니다. 리소스는 애플리케이션 로드 밸런서 또는 API Gateway를 참조할 ARN 수 있습니다.

```
aws waf-regional associate-web-acl \
    --web-acl-id a123fae4-b567-8e90-1234-5ab67ac8ca90 \
    --resource-arn 12cs345-67cd-890b-1cd2-c3a4567d89f1
```

자세한 내용은 AWS WAF 개발자 안내서의 웹 작업을 ACLs 참조하세요.

- 자세한 API 내용은 명령 참조AssociateWebAcl의 섹션을 참조하세요. AWS CLI

### `put-logging-configuration`

다음 코드 예시에서는 `put-logging-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 Kinesis Firehose 스트림을 ACL ARN 사용하여 웹에 대한 로깅 구성을 생성하려면 ARN

다음 `put-logging-configuration` 예제는 리전 의 WAF ALB/APIGateway 에 대한 로깅 구성을 보여줍니다`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
```

출력:

```
{
    "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"
        ]
    }
}
```

- 자세한 API 내용은 명령 참조PutLoggingConfiguration의 섹션을 참조하세요. AWS CLI

### update-byte-match-set

다음 코드 예시에서는 update-byte-match-set을 사용하는 방법을 보여 줍니다.

AWS CLI

바이트 일치 세트를 업데이트하려면

다음 update-byte-match-set 명령은 에서 ByteMatchTuple 객체(필터)를 삭제합니다ByteMatchSet. updates 값에 큰따옴표가 포함되어 있으므로 값을 작은따옴표로 묶어야 합니다.

```
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="
```

자세한 내용은 AWS WAF 개발자 안내서의 문자열 일치 조건 작업을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateByteMatchSet의 섹션을 참조하세요. AWS CLI

### update-ip-set

다음 코드 예시에서는 update-ip-set을 사용하는 방법을 보여 줍니다.

AWS CLI

IP 세트를 업데이트하려면

다음 update-ip-set 명령은 IPv4 주소로 IPSet를 업데이트하고 IPv6 주소를 삭제합니다. get-change-token 명령을 실행change-token하여 의 값을 가져옵니다. 업데이트 값에는 포함된 큰따옴표가 포함되어 있으므로 값을 작은따옴표로 묶어야 합니다.

```
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
```

또는 JSON 파일을 사용하여 입력을 지정할 수 있습니다. 예:

```
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
```

의 콘텐츠 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"
        }
    }
]
```

자세한 내용은 AWS WAF 개발자 안내서의 IP 일치 조건 작업을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateIpSet의 섹션을 참조하세요. AWS CLI

### update-rule

다음 코드 예시에서는 update-rule을 사용하는 방법을 보여 줍니다.

AWS CLI

규칙을 업데이트하려면

다음 update-rule 명령은 규칙에서 Predicate 객체를 삭제합니다. updates 값에 큰따옴표가 포함되어 있으므로 전체 값을 작은따옴표로 묶어야 합니다.

```
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"}'
```

자세한 내용은 AWS WAF 개발자 안내서의 규칙 작업을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateRule의 섹션을 참조하세요. AWS CLI

### update-size-constraint-set

다음 코드 예시에서는 update-size-constraint-set을 사용하는 방법을 보여 줍니다.

AWS CLI

크기 제약 조건 세트를 업데이트하려면

다음 update-size-constraint-set 명령은 크기 제약 조건 세트에서 SizeConstraint` 객체(필터)를 삭제합니다. updates 값에 포함된 큰따옴표가 포함되어 있으므로 전체 값을 작은따옴표로 묶어야 합니다.

```
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
```

자세한 내용은 AWS WAF 개발자 안내서의 크기 제약 조건 작업을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateSizeConstraintSet의 섹션을 참조하세요. AWS CLI

### update-sql-injection-match-set

다음 코드 예시에서는 update-sql-injection-match-set을 사용하는 방법을 보여 줍니다.

AWS CLI

SQL 주입 일치 세트를 업데이트하려면

다음 update-sql-injection-match-set 명령은 SQL 주입 일치 세트의 SqlInjectionMatchTuple 객체(필터)를 삭제합니다. updates 값에 포함된 큰따옴표가 포함되어 있으므로 전체 값을 작은따옴표로 묶어야 합니다. :

aws waf-regional update-sql-injection-match-set -sql-injection-match-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90 --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 --updates 'Action=DELETE',SqlInjectionMatchTuple={FieldToMatch={Type='QUERY_STRING'},TextTransformation

자세한 내용은 AWS WAF 개발자 안내서의 SQL 주입 일치 조건 작업을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateSqlInjectionMatchSet의 섹션을 참조하세요. AWS CLI

### update-web-acl

다음 코드 예시에서는 update-web-acl을 사용하는 방법을 보여 줍니다.

AWS CLI

웹을 업데이트하려면 ACL

다음 update-web-acl 명령은 웹 에서 ActivatedRule 객체를 삭제합니다ACL. updates 값에 포함된 큰따옴표가 포함되어 있으므로 전체 값을 작은따옴표로 묶어야 합니다.

```
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"}'
```

자세한 내용은 AWS WAF 개발자 안내서의 웹 작업을 ACLs 참조하세요.

- 자세한 API 내용은 명령 참조UpdateWebAcl의 섹션을 참조하세요. AWS CLI

**update-xss-match-set**

다음 코드 예시에서는 update-xss-match-set을 사용하는 방법을 보여 줍니다.

AWS CLI

를 업데이트하려면 XSSMatchSet

다음 update-xss-match-set 명령은 에서 XssMatchTuple 객체(필터)를 삭제합니다XssMatchSet. updates 값에 포함된 큰따옴표가 포함되어 있으므로 전체 값을 작은따옴표로 묶어야 합니다.

```
aws waf-regional update-xss-match-set \
    --xss-match-set-id a123fae4-b567-8e90-1234-5ab67ac8ca90 \
    --change-token 12cs345-67cd-890b-1cd2-c3a4567d89f1 \
    --updates
 'Action="DELETE",XssMatchTuple={FieldToMatch={Type="QUERY_STRING"},TextTransformation="URL_
```

자세한 내용은 AWS WAF 개발자 안내서의 교차 사이트 스크립팅 일치 조건 작업을 참조하세요.

- 자세한 API 내용은 명령 참조UpdateXssMatchSet의 섹션을 참조하세요. AWS CLI

## AWS WAFV2 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS WAFV2.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### `associate-web-acl`

다음 코드 예시에서는 `associate-web-acl`을 사용하는 방법을 보여 줍니다.

AWS CLI

웹을 리전 AWS 리소스ACL와 연결하려면

다음 `associate-web-acl` 예제에서는 지정된 웹을 Application Load Balancer ACL와 연결합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 , AWS WAF AWS 방화벽 관리자 및 AWS Shield 고급 개발자 안내서의 AWS 리소스ACL와 웹 연결 또는 연결 해제를 참조하세요.

- 자세한 API 내용은 명령 참조AssociateWebAcl의 섹션을 참조하세요. AWS CLI

### `check-capacity`

다음 코드 예시에서는 `check-capacity`을 사용하는 방법을 보여 줍니다.

AWS CLI

규칙 집합에서 사용하는 용량을 얻으려면

다음은 속도 기반 규칙 문이 포함된 규칙 세트와 중첩된 AND 규칙이 포함된 규칙 문의 용량 요구 사항을 `check-capacity` 검색합니다.

```
aws wafv2 check-capacity \
    --scope REGIONAL \
    --rules file://waf-rule-list.json \
    --region us-west-2
```

file://waf-rule-list.json의 내용:

```
[
    {
        "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"
        }
    }
]
```

출력:

```
{
    "Capacity":15
}
```

자세한 내용은 , AWS WAF AWS 방화벽 관리자 및 AWS Shield 고급 개발자 안내서의 AWS WAF 웹 ACL 용량 단위(WCU)를 참조하세요.

- 자세한 API 내용은 명령 참조CheckCapacity의 섹션을 참조하세요. AWS CLI

### create-ip-set

다음 코드 예시에서는 create-ip-set을 사용하는 방법을 보여 줍니다.

AWS CLI

웹 ACLs 및 규칙 그룹에서 사용할 IP 세트를 생성하려면

다음 `create-ip-set` 명령은 단일 주소 범위 사양으로 IP 세트를 생성합니다.

```
aws wafv2 create-ip-set \
    --name testip \
    --scope REGIONAL \
    --ip-address-version IPV4 \
    --addresses 198.51.100.0/16
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 , AWS WAF AWS 방화벽 관리자 및 AWS Shield 고급 개발자 안내서의 IP 세트 및 Regex 패턴 세트를 참조하세요.

- 자세한 API 내용은 명령 참조CreateIpSet의 섹션을 참조하세요. AWS CLI

### `create-regex-pattern-set`

다음 코드 예시에서는 `create-regex-pattern-set`을 사용하는 방법을 보여 줍니다.

AWS CLI

웹 ACLs 및 규칙 그룹에서 사용할 정규식 패턴 세트를 생성하려면

다음 `create-regex-pattern-set` 명령은 두 개의 정규식 패턴이 지정된 정규식 패턴 세트를 생성합니다.

```
aws wafv2 create-regex-pattern-set \
```

```
    --name regexPatterSet01 \
    --scope REGIONAL \
    --description 'Test web-acl' \
    --regular-expression-list '[{"RegexString": "/[0-9]*/"},{"RegexString": "/[a-
z]*/"}]'
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 , AWS WAF AWS 방화벽 관리자 및 AWS Shield 고급 개발자 안내서의 IP 세트 및 Regex 패턴 세트를 참조하세요.

- 자세한 API 내용은 명령 참조CreateRegexPatternSet의 섹션을 참조하세요. AWS CLI

### create-rule-group

다음 코드 예시에서는 create-rule-group을 사용하는 방법을 보여 줍니다.

AWS CLI

웹에서 사용할 사용자 지정 규칙 그룹을 생성하려면 ACLs

다음 create-rule-group 명령은 리전에서 사용할 사용자 지정 규칙 그룹을 생성합니다. 그룹의 규칙 문은 JSON형식이 지정된 파일로 제공됩니다.

```
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
```

file://waf-rule.json의 내용:

```
[
    {
        "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"
        }
    }
]
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 , AWS WAF AWS 방화벽 관리자 및 AWS Shield 고급 개발자 안내서의 자체 규칙 그룹 관리를 참조하세요.

- 자세한 API 내용은 명령 참조CreateRuleGroup의 섹션을 참조하세요. AWS CLI

### `create-web-acl`

다음 코드 예시에서는 `create-web-acl`을 사용하는 방법을 보여 줍니다.

AWS CLI

웹을 생성하려면 ACL

다음 `create-web-acl` 명령은 리전ACL에서 사용할 웹을 생성합니다. 웹에 대한 규칙 문ACL은 JSON형식이 지정된 파일로 제공됩니다.

```
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
```

file://waf-rule.json의 내용:

```
[
    {
        "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"
        }
    }
]
```

출력:

```
{
    "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"
    }
}
```

자세한 내용은 , 방화벽 관리자 및 Shield 고급 개발자 안내서의 웹 액세스 제어 목록(웹 ACL) 관리 및 사용을 참조하세요. AWS WAF AWS AWS

- 자세한 API 내용은 명령 참조CreateWebAcl의 섹션을 참조하세요. AWS CLI

### `delete-ip-set`

다음 코드 예시에서는 `delete-ip-set`을 사용하는 방법을 보여 줍니다.

AWS CLI

IP 세트를 삭제하려면

다음은 지정된 IP 세트를 `delete-ip-set` 삭제합니다. 이 호출에는 호출, 및 에서 얻을 수 있는 잠금 `list-ip-sets`토큰에서 얻을 수 있는 ID`list-ip-sets`가 필요합니다`get-ip-set`.

```
aws wafv2 delete-ip-set \
```

```
    --name test1 \
    --scope REGIONAL \
    --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --lock-token 46851772-db6f-459d-9385-49428812e357
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 , AWS WAF AWS 방화벽 관리자 AWS 및 Shield 고급 개발자 안내서의 IP 세트 및 Regex 패턴 세트를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteIpSet의 섹션을 참조하세요. AWS CLI

### delete-logging-configuration

다음 코드 예시에서는 delete-logging-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

웹에 대한 로깅을 비활성화하려면 ACL

다음은 지정된 웹 에서 로깅 구성을 delete-logging-configuration 제거합니다ACL.

```
aws wafv2 delete-logging-configuration \
    --resource-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/test/
a1b2c3d4-5678-90ab-cdef-EXAMPLE22222
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 , AWS WAF AWS 방화벽 관리자 및 AWS Shield 고급 개발자 안내서의 웹 ACL 트래픽 정보 로깅을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteLoggingConfiguration의 섹션을 참조하세요. AWS CLI

### delete-regex-pattern-set

다음 코드 예시에서는 delete-regex-pattern-set을 사용하는 방법을 보여 줍니다.

AWS CLI

정규식 패턴 세트를 삭제하려면

다음은 지정된 정규식 패턴 세트에 대한 설정을 delete-regex-pattern-set 업데이트합니다. 이 호출에는 호출, 에서 가져올 수 있는 IDlist-regex-pattern-sets와 호출 list-regex-

pattern-sets 또는 호출 에서 가져올 수 있는 잠금 토큰이 필요합니다get-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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 , AWS WAF AWS 방화벽 관리자 및 AWS Shield 고급 개발자 안내서의 IP 세트 및 Regex 패턴 세트를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteRegexPatternSet의 섹션을 참조하세요. AWS CLI

### delete-rule-group

다음 코드 예시에서는 delete-rule-group을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 규칙 그룹을 삭제하려면

다음은 지정된 사용자 지정 규칙 그룹을 delete-rule-group 삭제합니다. 이 호출에는 호출, 에서 가져올 수 있는 IDlist-rule-groups와 호출 list-rule-groups 또는 호출 에서 가져올 수 있는 잠금 토큰이 필요합니다get-rule-group.

```
aws wafv2 delete-rule-group \
    --name TestRuleGroup \
    --scope REGIONAL \
    --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --lock-token 7b3bcec2-0000-0000-0000-563bf47249f0
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 , AWS WAF AWS 방화벽 관리자 및 AWS Shield 고급 개발자 안내서의 자체 규칙 그룹 관리를 참조하세요.

- 자세한 API 내용은 명령 참조DeleteRuleGroup의 섹션을 참조하세요. AWS CLI

### delete-web-acl

다음 코드 예시에서는 delete-web-acl을 사용하는 방법을 보여 줍니다.

AWS CLI

웹을 삭제하려면 ACL

다음은 계정ACL에서 지정된 웹을 delete-web-acl 삭제합니다. 웹은 리소스와 연결되지 않은 경우에만 삭제할 ACL 수 있습니다. 이 호출에는 호출, 에서 가져올 수 있는 IDlist-web-acls와 호출 list-web-acls 또는 호출 에서 가져올 수 있는 잠금 토큰이 필요합니다get-web-acl.

```
aws wafv2 delete-web-acl \
    --name test \
    --scope REGIONAL \
    --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --lock-token ebab4ed2-155e-4c9a-9efb-e4c45665b1f5
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 , 방화벽 관리자 및 Shield 고급 개발자 안내서의 웹 액세스 제어 목록(웹 ACL) 관리 및 사용을 참조하세요. AWS WAF AWS AWS

- 자세한 API 내용은 명령 참조DeleteWebAcl의 섹션을 참조하세요. AWS CLI

### describe-managed-rule-group

다음 코드 예시에서는 describe-managed-rule-group을 사용하는 방법을 보여 줍니다.

AWS CLI

관리형 규칙 그룹에 대한 설명을 검색하려면

다음은 AWS 관리형 규칙 그룹에 대한 설명을 describe-managed-rule-group 검색합니다.

```
aws wafv2 describe-managed-rule-group \
    --vendor-name AWS \
    --name AWSManagedRulesCommonRuleSet \
    --scope REGIONAL
```

출력:

```
{
```

```
    "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_URIPATH",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "EC2MetaDataSSRF_BODY",
            "Action": {
                "Block": {}
            }
        },
```

```
        {
            "Name": "EC2MetaDataSSRF_COOKIE",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "EC2MetaDataSSRF_URIPATH",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "EC2MetaDataSSRF_QUERYARGUMENTS",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "GenericLFI_QUERYARGUMENTS",
            "Action": {
                "Block": {}
            }
        },
        {
            }
            "Name": "GenericLFI_URIPATH",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "GenericLFI_BODY",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "RestrictedExtensions_URIPATH",
            "Action": {
                "Block": {}
            }
        },
        {
```

```
            "Name": "RestrictedExtensions_QUERYARGUMENTS",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "GenericRFI_QUERYARGUMENTS",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "GenericRFI_BODY",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "GenericRFI_URIPATH",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "CrossSiteScripting_COOKIE",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "CrossSiteScripting_QUERYARGUMENTS",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "CrossSiteScripting_BODY",
            "Action": {
                "Block": {}
            }
        },
        {
            "Name": "CrossSiteScripting_URIPATH",
            "Action": {
```

```
                "Block": {}
            }
        }
    ]
}
```

자세한 내용은 , AWS WAF AWS 방화벽 관리자 및 AWS Shield 고급 개발자 안내서의 관리형 규칙 그룹을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeManagedRuleGroup의 섹션을 참조하세요. AWS CLI

### disassociate-web-acl

다음 코드 예시에서는 disassociate-web-acl을 사용하는 방법을 보여 줍니다.

AWS CLI

리전 AWS 리소스ACL에서 웹 연결을 해제하려면

다음 disassociate-web-acl 예제에서는 지정된 Application Load Balancer 에서 기존 웹 ACL 연결을 제거합니다.

```
aws wafv2 disassociate-web-acl \
    --resource-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/waf-cli-alb/1ea17125f8b25a2a \
    --region us-west-2
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 , AWS WAF AWS 방화벽 관리자 및 AWS Shield 고급 개발자 안내서의 AWS 리소스ACL와 웹 연결 또는 연결 해제를 참조하세요.

- 자세한 API 내용은 명령 참조DisassociateWebAcl의 섹션을 참조하세요. AWS CLI

### get-ip-set

다음 코드 예시에서는 get-ip-set을 사용하는 방법을 보여 줍니다.

AWS CLI

특정 IP 세트를 검색하려면

다음은 지정된 이름, 범위 및 ID가 있는 IP 세트를 get-ip-set 검색합니다. create-ip-set 및 명령에서 IP 세트의 ID를 가져올 수 있습니다list-ip-sets.

```
aws wafv2 get-ip-set \
    --name testip \
    --scope REGIONAL \
    --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

출력:

```
{
    "IPSet":{
        "Description":"",
        "Name":"testip",
        "IPAddressVersion":"IPV4",
        "Id":"a1b2c3d4-5678-90ab-cdef-EXAMPLE1111",
        "ARN":"arn:aws:wafv2:us-west-2:123456789012:regional/ipset/testip/
a1b2c3d4-5678-90ab-cdef-EXAMPLE1111",
        "Addresses":[
            "192.0.2.0/16"
        ]
    },
    "LockToken":"447e55ac-2396-4c6d-b9f9-86b67c17f8b5"
}
```

자세한 내용은 , AWS WAF AWS 방화벽 관리자 및 AWS Shield 고급 개발자 안내서의 IP 세트 및 Regex 패턴 세트를 참조하세요.

- 자세한 API 내용은 명령 참조GetIpSet의 섹션을 참조하세요. AWS CLI

### get-logging-configuration

다음 코드 예시에서는 get-logging-configuration을 사용하는 방법을 보여 줍니다.

AWS CLI

웹의 로깅 구성을 검색하려면 ACL

다음은 지정된 웹 에 대한 로깅 구성을 get-logging-configuration 검색합니다ACL.

```
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
```

출력:

```
{
    "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"
        ]
    }
}
```

자세한 내용은 , AWS WAF AWS 방화벽 관리자 및 AWS Shield 고급 개발자 안내서의 웹 ACL 트래픽 정보 로깅을 참조하세요.

- 자세한 API 내용은 명령 참조GetLoggingConfiguration의 섹션을 참조하세요. AWS CLI

### get-rate-based-statement-managed-keys

다음 코드 예시에서는 get-rate-based-statement-managed-keys을 사용하는 방법을 보여 줍니다.

AWS CLI

속도 기반 규칙에 의해 차단된 IP 주소 목록을 검색하려면

다음은 리전 애플리케이션에 사용 중인 속도 기반 규칙에 의해 현재 차단된 IP 주소를 get-rate-based-statement-managed-keys 검색합니다.

```
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
```

출력:

```
{
    "ManagedKeysIPV4":{
        "IPAddressVersion":"IPV4",
        "Addresses":[
            "198.51.100.0/32"
        ]
    },
    "ManagedKeysIPV6":{
        "IPAddressVersion":"IPV6",
        "Addresses":[

        ]
    }
}
```

자세한 내용은 ,AWS WAF AWS Firewall Manager 및 AWS Shield Advanced Developer Guide의 Rate-Based Rule Statement를 참조하세요.

- 자세한 API 내용은 명령 참조GetRateBasedStatementManagedKeys의 섹션을 참조하세요. AWS CLI

### get-regex-pattern-set

다음 코드 예시에서는 get-regex-pattern-set을 사용하는 방법을 보여 줍니다.

AWS CLI

특정 정규식 패턴 세트를 검색하려면

다음은 지정된 이름, 범위, 리전 및 ID가 있는 정규식 패턴 세트를 get-regex-pattern-set 검색합니다. create-regex-pattern-set 및 명령에서 정규식 패턴 세트의 ID를 가져올 수 있습니다list-regex-pattern-sets.

```
aws wafv2 get-regex-pattern-set \
    --name regexPatterSet01 \
```

```
    --scope REGIONAL \
    --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --region us-west-2
```

출력:

```
{
    "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"
}
```

자세한 내용은 , AWS WAF AWS 방화벽 관리자 및 AWS Shield 고급 개발자 안내서의 IP 세트 및 Regex 패턴 세트를 참조하세요.

- 자세한 API 내용은 명령 참조GetRegexPatternSet의 섹션을 참조하세요. AWS CLI

### get-rule-group

다음 코드 예시에서는 get-rule-group을 사용하는 방법을 보여 줍니다.

AWS CLI

특정 사용자 지정 규칙 그룹을 검색하려면

다음은 지정된 이름, 범위 및 ID로 사용자 지정 규칙 그룹을 get-rule-group 검색합니다. create-rule-group 및 명령에서 규칙 그룹의 ID를 가져올 수 있습니다list-rule-groups.

```
aws wafv2 get-rule-group \
```

```
    --name ff \
    --scope REGIONAL \
    --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

출력:

```
{
    "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"
}
```

자세한 내용은 , AWS WAF AWS 방화벽 관리자 및 AWS Shield 고급 개발자 안내서의 자체 규칙 그룹 관리를 참조하세요.

- 자세한 API 내용은 명령 참조GetRuleGroup의 섹션을 참조하세요. AWS CLI

### get-sampled-requests

다음 코드 예시에서는 get-sampled-requests을 사용하는 방법을 보여 줍니다.

AWS CLI

웹에 대한 웹 요청 샘플을 검색하려면 ACL

다음은 지정된 웹 ACL, 규칙 지표 및 기간에 대한 샘플링된 웹 요청을 get-sampled-requests 검색합니다.

```
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
```

출력:

```
{
    "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
}
```

자세한 내용은 , AWS WAF AWS 방화벽 관리자 및 AWS Shield 고급 개발자 안내서의 웹 요청 샘플 보기를 참조하세요.

- 자세한 API 내용은 명령 참조GetSampledRequests의 섹션을 참조하세요. AWS CLI

### get-web-acl-for-resource

다음 코드 예시에서는 get-web-acl-for-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

AWS 리소스와 ACL 연결된 웹을 검색하려면

다음은 지정된 리소스와 ACL 연결된 웹JSON의 를 get-web-acl-for-resource 검색합니다.

```
aws wafv2 get-web-acl-for-resource \
    --resource-arn arn:aws:elasticloadbalancing:us-west-2:123456789012:loadbalancer/app/waf-cli-alb/1ea17125f8b25a2a
```

출력:

```
{
    "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":"dGVzdHN0cmluZw==",
                                    "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"
    }
}
```

자세한 내용은 , AWS WAF AWS 방화벽 관리자 및 AWS Shield 고급 개발자 안내서의 AWS 리소스ACL와 웹 연결 또는 연결 해제를 참조하세요.

- 자세한 API 내용은 명령 참조GetWebAclForResource의 섹션을 참조하세요. AWS CLI

### `get-web-acl`

다음 코드 예시에서는 `get-web-acl`을 사용하는 방법을 보여 줍니다.

AWS CLI

웹을 검색하려면 ACL

다음은 지정된 이름, 범위 및 IDACL로 웹을 `get-web-acl` 검색합니다. `create-web-acl` 및 명령ACL에서 웹의 ID를 가져올 수 있습니다`list-web-acls`.

```
aws wafv2 get-web-acl \
    --name test01 \
```

```
    --scope REGIONAL \
    --id a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

출력:

```
{
    "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":"dGVzdHN0cmluZw==",
                                    "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"
}
```

자세한 내용은 , 방화벽 관리자 및 Shield 고급 개발자 안내서의 웹 액세스 제어 목록(웹 ACL) 관리 및 사용을 참조하세요. AWS WAF AWS AWS

- 자세한 API 내용은 명령 참조GetWebAcl의 섹션을 참조하세요. AWS CLI

### list-available-managed-rule-groups

다음 코드 예시에서는 list-available-managed-rule-groups을 사용하는 방법을 보여 줍니다.

AWS CLI

관리형 규칙 그룹을 검색하려면

다음은 현재 웹 에서 사용할 수 있는 모든 관리형 규칙 그룹의 목록을 list-available-managed-rule-groups 반환합니다ACLs.

```
aws wafv2 list-available-managed-rule-groups \
    --scope REGIONAL
```

출력:

```
{
   "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."
        }
    ]
}
```

자세한 내용은 , AWS WAF AWS 방화벽 관리자 및 AWS Shield 고급 개발자 안내서의 관리형 규칙 그룹을 참조하세요.

- 자세한 API 내용은 명령 참조ListAvailableManagedRuleGroups의 섹션을 참조하세요. AWS CLI

### `list-ip-sets`

다음 코드 예시에서는 `list-ip-sets`을 사용하는 방법을 보여 줍니다.

AWS CLI

IP 세트 목록을 검색하려면

다음은 리전 범위가 있는 계정의 모든 IP 세트를 `list-ip-sets` 검색합니다.

```
aws wafv2 list-ip-sets \
    --scope REGIONAL
```

출력:

```
{
    "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"
}
```

자세한 내용은 , AWS WAF AWS 방화벽 관리자 및 AWS Shield 고급 개발자 안내서의 IP 세트 및 Regex 패턴 세트를 참조하세요.

- 자세한 API 내용은 명령 참조ListIpSets의 섹션을 참조하세요. AWS CLI

### `list-logging-configurations`

다음 코드 예시에서는 `list-logging-configurations`을 사용하는 방법을 보여 줍니다.

AWS CLI

리전의 모든 로깅 구성 목록을 검색하려면

다음은 `us-west-2` 리전에서 리전별로 사용하도록 ACLs 범위가 지정된 웹의 모든 로깅 구성을 `list-logging-configurations` 검색합니다.

```
aws wafv2 list-logging-configurations \
    --scope REGIONAL \
    --region us-west-2
```

출력:

```
{
    "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"
            ]
        }
    ]
}
```

자세한 내용은 , AWS WAF AWS 방화벽 관리자 및 AWS Shield 고급 개발자 안내서의 웹 ACL 트래픽 정보 로깅을 참조하세요.

- 자세한 API 내용은 명령 참조ListLoggingConfigurations의 섹션을 참조하세요. AWS CLI

### `list-regex-pattern-sets`

다음 코드 예시에서는 `list-regex-pattern-sets`을 사용하는 방법을 보여 줍니다.

AWS CLI

정규식 패턴 세트 목록을 검색하려면

다음은 리전 에 정의된 계정의 모든 정규식 패턴 세트를 `list-regex-pattern-sets` 검색합니다`us-west-2`.

```
aws wafv2 list-regex-pattern-sets \
--scope REGIONAL \
--region us-west-2
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 , AWS WAF AWS 방화벽 관리자 및 AWS Shield 고급 개발자 안내서의 IP 세트 및 Regex 패턴 세트를 참조하세요.

- 자세한 API 내용은 명령 참조ListRegexPatternSets의 섹션을 참조하세요. AWS CLI

### `list-resources-for-web-acl`

다음 코드 예시에서는 `list-resources-for-web-acl`을 사용하는 방법을 보여 줍니다.

AWS CLI

웹과 연결된 리소스를 검색하려면 ACL

다음은 ACL 리전 에서 현재 지정된 웹과 연결된 API Gateway REST API 리소스를 `list-resources-for-web-acl` 검색합니다`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
```

출력:

```
{
    "ResourceArns": [
        "arn:aws:apigateway:us-west-2::/restapis/EXAMPLE111/stages/testing"
    ]
}
```

자세한 내용은 , AWS WAF AWS 방화벽 관리자 및 AWS Shield 고급 개발자 안내서의 AWS 리소스ACL와 웹 연결 또는 연결 해제를 참조하세요.

- 자세한 API 내용은 명령 참조ListResourcesForWebAcl의 섹션을 참조하세요. AWS CLI

### `list-rule-groups`

다음 코드 예시에서는 `list-rule-groups`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 규칙 그룹 목록을 검색하려면

다음은 지정된 범위 및 리전 위치의 계정에 정의된 모든 사용자 지정 규칙 그룹을 `list-rule-groups` 검색합니다.

```
aws wafv2 list-rule-groups \
    --scope REGIONAL \
    --region us-west-2
```

출력:

```
{
    "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"
```

```
}
```

자세한 내용은 , AWS WAF AWS 방화벽 관리자 및 AWS Shield 고급 개발자 안내서의 자체 규칙 그룹 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ListRuleGroups의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스의 AWS WAF 모든 태그를 검색하려면

다음은 지정된 웹 의 모든 태그 키, 값 페어 목록을 `list-tags-for-resource` 검색합니다ACL.

```
aws wafv2 list-tags-for-resource \
    --resource-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/testwebacl2/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

출력:

```
{
    "NextMarker":"",
    "TagInfoForResource":{
        "ResourceARN":"arn:aws:wafv2:us-west-2:123456789012:regional/webacl/
testwebacl2/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "TagList":[

        ]
    }
}
```

자세한 내용은 , AWS WAF AWS 방화벽 관리자 및 AWS Shield 고급 개발자 안내서의 시작하기를 참조하세요 AWS WAF.

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### `list-web-acls`

다음 코드 예시에서는 `list-web-acls`을 사용하는 방법을 보여 줍니다.

AWS CLI

범위에 ACLs 대한 웹을 검색하려면

다음은 지정된 범위의 계정에 ACLs 정의된 모든 웹을 `list-web-acls` 검색합니다.

```
aws wafv2 list-web-acls \
    --scope REGIONAL
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 ,Firewall Manager 및 Shield Advanced Developer Guide의 웹 액세스 제어 목록(웹 ACL) 관리 및 사용을 참조하세요. AWS WAF AWS AWS

- 자세한 API 내용은 명령 참조ListWebAcls의 섹션을 참조하세요. AWS CLI

### `put-logging-configuration`

다음 코드 예시에서는 `put-logging-configuration`을 사용하는 방법을 보여 줍니다.

AWS CLI

웹에 로깅 구성을 추가하려면 ACL

다음은 Amazon Kinesis Data Firehose 로깅 구성을 지정된 웹 `aws-waf-logs-custom-transformation`에 `put-logging-configuration` 추가하며 로그에서 필드가 ACL수정되지 않습니다.

```
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
```

출력:

```
{
    "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"
        ]
    }
}
```

자세한 내용은 , AWS WAF AWS 방화벽 관리자 및 AWS Shield 고급 개발자 안내서의 웹 ACL 트래픽 정보 로깅을 참조하세요.

- 자세한 API 내용은 명령 참조PutLoggingConfiguration의 섹션을 참조하세요. AWS CLI

### tag-resource

다음 코드 예시에서는 tag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 AWS WAF 태그를 추가하려면

다음 tag-resource 예제에서는 Name 및 값이 로 설정된 태그를 지정된 웹 AWSWAF에 추가합니다ACL.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 , AWS WAF AWS 방화벽 관리자 및 AWS Shield 고급 개발자 안내서의 시작하기를 참조하세요 AWS WAF.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### untag-resource

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에서 AWS WAF 태그를 제거하려면

다음 untag-resource 예제에서는 지정된 웹 KeyName에서 키가 있는 태그를 제거합니다ACL.

```
aws wafv2 untag-resource \
    --resource-arn arn:aws:wafv2:us-west-2:123456789012:regional/webacl/
apiGatewayWebAcl/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111 \
    --tag-keys "KeyName"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 , AWS WAF AWS 방화벽 관리자 및 AWS Shield 고급 개발자 안내서의 시작하기를 참조하세요 AWS WAF.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-ip-set

다음 코드 예시에서는 update-ip-set을 사용하는 방법을 보여 줍니다.

AWS CLI

기존 IP 세트에 대한 설정을 수정하려면

다음은 지정된 IP 세트에 대한 설정을 update-ip-set 업데이트합니다. 이 호출에는 호출, 및 에서 얻을 수 있는 list-ip-sets잠금 토큰에서 얻을 수 있는 IDlist-ip-sets가 필요합니다get-ip-set. 이 호출은 후속 업데이트에 사용할 수 있는 잠금 토큰도 반환합니다.

```
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
```

출력:

```
{
    "NextLockToken": "0674c84b-0304-47fe-8728-c6bff46af8fc"
}
```

자세한 내용은 , AWS WAF AWS 방화벽 관리자 및 AWS Shield 고급 개발자 안내서의 IP 세트 및 Regex 패턴 세트를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateIpSet의 섹션을 참조하세요. AWS CLI

### update-regex-pattern-set

다음 코드 예시에서는 update-regex-pattern-set을 사용하는 방법을 보여 줍니다.

AWS CLI

기존 정규식 패턴 세트의 설정을 수정하려면

다음은 지정된 정규식 패턴 세트에 대한 설정을 update-regex-pattern-set 업데이트합니다. 이 호출에는 호출에서 가져올 수 있는 ID, list-regex-pattern-sets및 호출에서 가져올 수 있는 잠금 토큰 list-regex-pattern-sets 및 가 필요합니다get-regex-pattern-set. 이 호출은 후속 업데이트에 사용할 수 있는 잠금 토큰도 반환합니다.

```
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
```

출력:

```
{
    "NextLockToken": "12ebc73e-fa68-417d-a9b8-2bdd761a4fa5"
}
```

자세한 내용은 , AWS WAF AWS 방화벽 관리자 및 AWS Shield 고급 개발자 안내서의 IP 세트 및 Regex 패턴 세트를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateRegexPatternSet의 섹션을 참조하세요. AWS CLI

### update-rule-group

다음 코드 예시에서는 update-rule-group을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 규칙 그룹을 업데이트하려면

다음은 기존 사용자 지정 규칙 그룹의 가시성 구성을 update-rule-group 변경합니다. 이 호출에는 호출, 및 에서 얻을 수 있는 list-rule-groups잠금 토큰에서 얻을 수 있는 IDlist-rule-groups가 필요합니다get-rule-group. 이 호출은 후속 업데이트에 사용할 수 있는 잠금 토큰도 반환합니다.

```
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
```

출력:

```
{
    "NextLockToken": "1eb5ec48-0000-0000-0000-ee9b906c541e"
}
```

자세한 내용은 , AWS WAF AWS 방화벽 관리자 및 AWS Shield 고급 개발자 안내서의 자체 규칙 그룹 관리를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateRuleGroup의 섹션을 참조하세요. AWS CLI

### update-web-acl

다음 코드 예시에서는 update-web-acl을 사용하는 방법을 보여 줍니다.

AWS CLI

웹을 업데이트하려면 ACL

다음은 기존 웹 에 대한 설정을 update-web-acl 변경합니다ACL. 이 호출에는 호출, 및 list-web-acls잠금 토큰에서 얻을 수 있는 ID와 호출에서 얻을 수 있는 기타 설정이 필요합니다get-web-acl. 이 호출은 후속 업데이트에 사용할 수 있는 잠금 토큰도 반환합니다.

```
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
```

출력:

```
{
    "NextLockToken": "714a0cfb-0000-0000-0000-2959c8b9a684"
}
```

자세한 내용은 ,Firewall Manager 및 Shield 고급 개발자 안내서의 웹 액세스 제어 목록(웹 ACL) 관리 및 사용을 참조하세요. AWS WAF AWS AWS

- 자세한 API 내용은 명령 참조UpdateWebAcl의 섹션을 참조하세요. AWS CLI

## 를 사용한 Amazon WorkDocs 예제 AWS CLI

다음 코드 예제에서는 Amazon 와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 WorkDocs.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 작업

### `abort-document-version-upload`

다음 코드 예시에서는 `abort-document-version-upload`을 사용하는 방법을 보여 줍니다.

AWS CLI

문서 버전 업로드를 중지하려면

이 예제는 이전에 시작된 문서 버전 업로드를 중지합니다.

명령:

```
aws workdocs abort-document-version-upload --document-
id feaba64d4efdf271c2521b60a2a44a8f057e84beaabbe22f012673132098 35f2 --version-
id 1536773972914-ddb67663e782e7ce8455ebc962217cf9f9e47b5a9a702e5c84dcccd417da9313
```

출력:

```
None
```

- 자세한 API 내용은 명령 참조AbortDocumentVersionUpload의 섹션을 참조하세요. AWS CLI

### `activate-user`

다음 코드 예시에서는 `activate-user`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자를 활성화하려면

이 예제에서는 비활성 사용자를 활성화합니다.

명령:

```
aws workdocs activate-user --user-
id "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c"
```

출력:

```
{
  "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"
          }
      }
  }
}
```

- 자세한 API 내용은 명령 참조ActivateUser의 섹션을 참조하세요. AWS CLI

### add-resource-permissions

다음 코드 예시에서는 add-resource-permissions을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 대한 권한을 추가하려면

이 예제에서는 지정된 보안 주체에 대한 권한을 리소스에 추가합니다.

명령:

```
aws workdocs add-resource-permissions --resource-
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --
principals Id=anonymous,Type=ANONYMOUS,Role=VIEWER
```

출력:

```
{
  "ShareResults": [
      {
          "PrincipalId": "anonymous",
          "Role": "VIEWER",
          "Status": "SUCCESS",
          "ShareId":
 "d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65",
          "StatusMessage": ""
      }
  ]
}
```

- 자세한 API 내용은 명령 참조AddResourcePermissions의 섹션을 참조하세요. AWS CLI

### `create-comment`

다음 코드 예시에서는 `create-comment`을 사용하는 방법을 보여 줍니다.

AWS CLI

새 주석을 추가하려면

이 예제에서는 지정된 문서 버전에 새 주석을 추가합니다.

명령:

```
aws workdocs create-comment --document-
id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --version-
id 1521672507741-9f7df0ea5dd0b121c4f3564a0c7c0b4da95cd12c635d3c442af337a88e297920 --
text "This is a comment."
```

출력:

```
{
  "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"
  }
}
```

- 자세한 API 내용은 명령 참조CreateComment의 섹션을 참조하세요. AWS CLI

### `create-custom-metadata`

다음 코드 예시에서는 `create-custom-metadata`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 지정 메타데이터를 생성하려면

이 예제에서는 지정된 문서에 대한 사용자 지정 메타데이터를 생성합니다.

명령:

```
aws workdocs create-custom-metadata --resource-
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --custom-
metadata KeyName1=example,KeyName2=example2
```

출력:

```
None
```

- 자세한 API 내용은 명령 참조CreateCustomMetadata의 섹션을 참조하세요. AWS CLI

### create-folder

다음 코드 예시에서는 create-folder을 사용하는 방법을 보여 줍니다.

AWS CLI

폴더를 생성하려면

이 예제에서는 폴더를 생성합니다.

명령:

```
aws workdocs create-folder --name documents --parent-folder-
id 1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678
```

출력:

```
{
  "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
  }
}
```

- 자세한 API 내용은 명령 참조CreateFolder의 섹션을 참조하세요. AWS CLI

### create-labels

다음 코드 예시에서는 create-labels을 사용하는 방법을 보여 줍니다.

AWS CLI

레이블을 생성하려면

이 예제에서는 문서에 대한 일련의 레이블을 생성합니다.

명령:

```
aws workdocs create-labels --resource-
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --
labels "documents" "examples" "my_documents"
```

출력:

```
None
```

- 자세한 API 내용은 명령 참조CreateLabels의 섹션을 참조하세요. AWS CLI

### `create-notification-subscription`

다음 코드 예시에서는 `create-notification-subscription`을 사용하는 방법을 보여 줍니다.

AWS CLI

알림 구독을 생성하려면

다음 `create-notification-subscription` 예제에서는 지정된 Amazon WorkDocs 조직에 대한 알림 구독을 구성합니다.

```
aws workdocs create-notification-subscription \
    --organization-id d-123456789c \
    --protocol HTTPS \
    --subscription-type ALL \
    --notification-endpoint "https://example.com/example"
```

출력:

```
{
    "Subscription": {
        "SubscriptionId": "123ab4c5-678d-901e-f23g-45h6789j0123",
        "EndPoint": "https://example.com/example",
        "Protocol": "HTTPS"
```

```
    }
}
```

자세한 내용은 Amazon WorkDocs 개발자 안내서의 알림 구독을 참조하세요.

- 자세한 API 내용은 명령 참조CreateNotificationSubscription의 섹션을 참조하세요. AWS CLI

### create-user

다음 코드 예시에서는 create-user을 사용하는 방법을 보여 줍니다.

AWS CLI

새 사용자를 생성하려면

이 예제에서는 Simple AD 또는 Microsoft AD 디렉터리에 새 사용자를 생성합니다.

명령:

```
aws workdocs create-user --organization-id d-926726012c --username exampleUser2
 --email-address exampleUser2@site.awsapps.com --given-name example2Name --
surname example2Surname --password examplePa$$w0rd
```

출력:

```
{
  "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"
            }
        }
    }
}
```

- 자세한 API 내용은 명령 참조CreateUser의 섹션을 참조하세요. AWS CLI

### `deactivate-user`

다음 코드 예시에서는 `deactivate-user`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자를 비활성화하려면

이 예제에서는 활성 사용자를 비활성화합니다.

명령:

```
aws workdocs deactivate-user --user-
id "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c"
```

출력:

```
None
```

- 자세한 API 내용은 명령 참조DeactivateUser의 섹션을 참조하세요. AWS CLI

### `delete-comment`

다음 코드 예시에서는 `delete-comment`을 사용하는 방법을 보여 줍니다.

AWS CLI

문서 버전에서 지정된 주석을 삭제하려면

이 예제에서는 지정된 문서 버전에서 지정된 주석을 삭제합니다.

명령:

```
aws workdocs delete-comment --document-
id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --version-
id 1521672507741-9f7df0ea5dd0b121c4f3564a0c7c0b4da95cd12c635d3c442af337a88e297920 --
comment-id 1534799058197-
c7f5c84de9115875bbca93e0367bbebac609541d461636b760849b88b1609dd5
```

출력:

```
None
```

- 자세한 API 내용은 명령 참조DeleteComment의 섹션을 참조하세요. AWS CLI

### `delete-custom-metadata`

다음 코드 예시에서는 `delete-custom-metadata`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에서 사용자 지정 메타데이터를 삭제하려면

이 예제에서는 지정된 리소스에서 모든 사용자 지정 메타데이터를 삭제합니다.

명령:

```
aws workdocs delete-custom-metadata --resource-
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --delete-all
```

출력:

```
None
```

- 자세한 API 내용은 명령 참조DeleteCustomMetadata의 섹션을 참조하세요. AWS CLI

### `delete-document`

다음 코드 예시에서는 `delete-document`을 사용하는 방법을 보여 줍니다.

AWS CLI

문서를 삭제하는 방법

이 예제에서는 지정된 문서를 삭제합니다.

명령:

```
aws workdocs delete-document --document-
id b83ed5e5b167b65ef69de9d597627ff1a0d4f07a45e67f1fab7d26b54427de0a
```

출력:

```
None
```

- 자세한 API 내용은 명령 참조DeleteDocument의 섹션을 참조하세요. AWS CLI

### `delete-folder-contents`

다음 코드 예시에서는 `delete-folder-contents`을 사용하는 방법을 보여 줍니다.

AWS CLI

폴더의 내용을 삭제하려면

이 예제에서는 지정된 폴더의 내용을 삭제합니다.

명령:

```
aws workdocs delete-folder-contents --folder-
id 26fa8aa4ba2071447c194f7b150b07149dbdb9e1c8a301872dcd93a4735ce65d
```

출력:

```
None
```

- 자세한 API 내용은 명령 참조DeleteFolderContents의 섹션을 참조하세요. AWS CLI

### `delete-folder`

다음 코드 예시에서는 `delete-folder`을 사용하는 방법을 보여 줍니다.

AWS CLI

폴더를 삭제하려면

이 예제에서는 지정된 폴더를 삭제합니다.

명령:

```
aws workdocs delete-folder --folder-
id 26fa8aa4ba2071447c194f7b150b07149dbdb9e1c8a301872dcd93a4735ce65d
```

출력:

```
None
```

- 자세한 API 내용은 명령 참조DeleteFolder의 섹션을 참조하세요. AWS CLI

### `delete-labels`

다음 코드 예시에서는 `delete-labels`을 사용하는 방법을 보여 줍니다.

AWS CLI

레이블을 삭제하려면

이 예제에서는 문서에서 지정된 레이블을 삭제합니다.

명령:

```
aws workdocs delete-labels --resource-
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --
labels "documents" "examples"
```

출력:

```
None
```

- 자세한 API 내용은 명령 참조DeleteLabels의 섹션을 참조하세요. AWS CLI

### `delete-notification-subscription`

다음 코드 예시에서는 `delete-notification-subscription`을 사용하는 방법을 보여 줍니다.

AWS CLI

알림 구독을 삭제하려면

다음 delete-notification-subscription 예제에서는 지정된 알림 구독을 삭제합니다.

```
aws workdocs delete-notification-subscription \
    --subscription-id 123ab4c5-678d-901e-f23g-45h6789j0123 \
    --organization-id d-123456789c
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon WorkDocs 개발자 안내서의 알림 구독을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteNotificationSubscription의 섹션을 참조하세요. AWS CLI

### delete-user

다음 코드 예시에서는 delete-user을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 삭제

이 예시는 사용자를 삭제합니다.

명령:

```
aws workdocs delete-user --user-
id "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c"
```

출력:

```
None
```

- 자세한 API 내용은 명령 참조DeleteUser의 섹션을 참조하세요. AWS CLI

### describe-activities

다음 코드 예시에서는 describe-activities을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 활동 목록을 가져오려면

이 예제에서는 지정된 조직에 대한 최신 사용자 활동 목록을 반환하고, 최근 두 활동에 대한 제한이 설정됩니다.

명령:

```
aws workdocs describe-activities --organization-id d-926726012c --limit 2
```

출력:

```
{
  "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":
 "DnF1ZXJ5VGhlbkZldGNoAgAAAAAAAAS7FmlTaU1OdlFTU1h1UU00VVFIbDlRWHcAAAAAAAAJTRY3bWh5eUgzaVF1ZX
}
```

- 자세한 API 내용은 명령 참조DescribeActivities의 섹션을 참조하세요. AWS CLI

### describe-comments

다음 코드 예시에서는 describe-comments을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 문서 버전에 대한 모든 설명을 나열하려면

이 예제에서는 지정된 문서 버전에 대한 모든 설명을 나열합니다.

명령:

```
aws workdocs describe-comments --document-
id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --version-
id 1521672507741-9f7df0ea5dd0b121c4f3564a0c7c0b4da95cd12c635d3c442af337a88e297920
```

출력:

```
{
  "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeComments의 섹션을 참조하세요. AWS CLI

### `describe-document-versions`

다음 코드 예시에서는 `describe-document-versions`을 사용하는 방법을 보여 줍니다.

AWS CLI

문서의 버전을 검색하려면

이 예제에서는 초기화된 버전과 URL 소스 문서의 를 포함하여 지정된 문서의 문서 버전을 검색합니다.

명령:

```
aws workdocs describe-document-versions --document-
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --fields SOURCE
```

출력:

```
{
    "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/
d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65/1534452029587-15e129dfc1875
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"
          }
      }
  ]
}
```

- 자세한 API 내용은 명령 참조DescribeDocumentVersions의 섹션을 참조하세요. AWS CLI

### describe-folder-contents

다음 코드 예시에서는 describe-folder-contents을 사용하는 방법을 보여 줍니다.

AWS CLI

폴더의 내용을 설명하려면

이 예제에서는 문서 및 하위 폴더를 포함하여 지정된 폴더의 모든 활성 내용을 오름차순으로 정렬하여 설명합니다.

명령:

```
aws workdocs describe-folder-contents --folder-
id 1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678 --sort DATE --
order ASCENDING --type ALL
```

출력:

```
{
  "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeFolderContents의 섹션을 참조하세요. AWS CLI

### `describe-groups`

다음 코드 예시에서는 `describe-groups`을 사용하는 방법을 보여 줍니다.

AWS CLI

그룹 목록을 검색하려면

다음 `describe-groups` 예제에서는 지정된 Amazon WorkDocs 조직과 연결된 그룹을 나열합니다.

```
aws workdocs describe-groups \
    --search-query "e" \
    --organization-id d-123456789c
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 Amazon 관리 안내서의 Amazon 시작하기 WorkDocs를 참조하세요. WorkDocs

- 자세한 API 내용은 명령 참조DescribeGroups의 섹션을 참조하세요. AWS CLI

### `describe-notification-subscriptions`

다음 코드 예시에서는 `describe-notification-subscriptions`을 사용하는 방법을 보여 줍니다.

AWS CLI

알림 구독 목록을 검색하려면

다음 `describe-notification-subscriptions` 예제에서는 지정된 Amazon WorkDocs 조직의 알림 구독을 검색합니다.

```
aws workdocs describe-notification-subscriptions \
    --organization-id d-123456789c
```

출력:

```
{
    "Subscriptions": [
        {
            "SubscriptionId": "123ab4c5-678d-901e-f23g-45h6789j0123",
            "EndPoint": "https://example.com/example",
            "Protocol": "HTTPS"
        }
    ]
```

```
}
```

자세한 내용은 Amazon WorkDocs 개발자 안내서의 알림 구독을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeNotificationSubscriptions의 섹션을 참조하세요. AWS CLI

### `describe-resource-permissions`

다음 코드 예시에서는 `describe-resource-permissions`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 대한 권한 목록을 가져오려면

이 예제에서는 지정된 리소스(문서 또는 폴더)에 대한 권한 목록을 반환합니다.

명령:

```
aws workdocs describe-resource-permissions --resource-
id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3
```

출력:

```
{
  "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"
                }
            ]
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeResourcePermissions의 섹션을 참조하세요. AWS CLI

### describe-users

다음 코드 예시에서는 describe-users을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 사용자의 세부 정보를 검색하려면

이 예제에서는 지정된 조직의 모든 사용자에 대한 세부 정보를 검색합니다.

명령:

```
aws workdocs describe-users --organization-id d-926726012c
```

출력:

```
{
    "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
        }
    ]
}
```

- 자세한 API 내용은 명령 참조DescribeUsers의 섹션을 참조하세요. AWS CLI

### `get-document-path`

다음 코드 예시에서는 `get-document-path`을 사용하는 방법을 보여 줍니다.

AWS CLI

문서의 경로 정보를 검색하려면

이 예제에서는 지정된 문서의 경로 정보(루트 폴더의 계층 구조)를 검색하고 상위 폴더의 이름을 포함합니다.

명령:

```
aws workdocs get-document-path --document-
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65 --fields NAME
```

출력:

```
{
  "Path": {
      "Components": [
          {
              "Id":
 "a43d29cbb8e7c4d25cfee8b803a504b0dc63e760b55ad0c611c6b87691eb6ff3",
              "Name": "/"
          },
          {
              "Id":
 "1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678",
              "Name": "Top Level Folder"
          },
          {
              "Id":
 "d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65",
              "Name": "exampleDoc.docx"
          }
      ]
  }
}
```

- 자세한 API 내용은 명령 참조GetDocumentPath의 섹션을 참조하세요. AWS CLI

### `get-document-version`

다음 코드 예시에서는 `get-document-version`을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 문서의 버전 메타데이터를 검색하려면

이 예제에서는 소스 URL 및 사용자 지정 메타데이터를 포함하여 지정된 문서의 버전 메타데이터를 검색합니다.

명령:

```
aws workdocs get-document-version --document-
id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --version-
id 1521672507741-9f7df0ea5dd0b121c4f3564a0c7c0b4da95cd12c635d3c442af337a88e297920 --
fields SOURCE --include-custom-metadata
```

출력:

```
{
  "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/15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3/152167
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=01Ab2c34d567e8f90123g456hi78j901k2345678l901234mno56pqr78EXAMPLE"
      }
  }
}
```

- 자세한 API 내용은 명령 참조GetDocumentVersion의 섹션을 참조하세요. AWS CLI

### `get-document`

다음 코드 예시에서는 `get-document`을 사용하는 방법을 보여 줍니다.

AWS CLI

문서 세부 정보를 검색하려면

이 예제에서는 지정된 문서의 세부 정보를 검색합니다.

명령:

```
aws workdocs get-document --document-
id d90d93c1fe44bad0c8471e973ebaab339090401a95e777cffa58e977d2983b65
```

출력:

```
{
  "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"
  }
}
```

- 자세한 API 내용은 명령 참조GetDocument의 섹션을 참조하세요. AWS CLI

### get-folder-path

다음 코드 예시에서는 get-folder-path을 사용하는 방법을 보여 줍니다.

AWS CLI

폴더의 경로 정보를 검색하려면

이 예제에서는 지정된 폴더에 대한 경로 정보(루트 폴더의 계층 구조)를 검색하고 상위 폴더의 이름을 포함합니다.

명령:

```
aws workdocs get-folder-path --folder-
id 50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08 --fields NAME
```

출력:

```
{
  "Path": {
      "Components": [
          {
              "Id":
 "a43d29cbb8e7c4d25cfee8b803a504b0dc63e760b55ad0c611c6b87691eb6ff3",
              "Name": "/"
          },
          {
              "Id":
 "1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678",
              "Name": "Top Level Folder"
          },
          {
              "Id":
 "50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08",
              "Name": "Sublevel Folder"
          }
      ]
  }
}
```

- 자세한 API 내용은 명령 참조GetFolderPath의 섹션을 참조하세요. AWS CLI

### `get-folder`

다음 코드 예시에서는 `get-folder`을 사용하는 방법을 보여 줍니다.

AWS CLI

폴더의 메타데이터를 검색하려면

이 예제에서는 지정된 폴더의 메타데이터를 검색합니다.

명령:

```
aws workdocs get-folder --folder-
id 50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08
```

출력:

```
{
  "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
  }
}
```

- 자세한 API 내용은 명령 참조GetFolder의 섹션을 참조하세요. AWS CLI

### get-resources

다음 코드 예시에서는 get-resources을 사용하는 방법을 보여 줍니다.

AWS CLI

공유 리소스를 검색하려면

다음 get-resources 예제에서는 지정된 Amazon WorkDocs 사용자와 공유된 리소스를 검색합니다.

```
aws workdocs get-resources \
    --user-id "S-1-1-11-1111111111-2222222222-3333333333-3333" \
    --collection-type SHARED_WITH_ME
```

출력:

```
{
```

```
"Folders": [],
"Documents": []
}
```

자세한 내용은 Amazon WorkDocs 사용 설명서의 파일 및 폴더 공유를 참조하세요.

- 자세한 API 내용은 명령 참조GetResources의 섹션을 참조하세요. AWS CLI

### initiate-document-version-upload

다음 코드 예시에서는 initiate-document-version-upload을 사용하는 방법을 보여 줍니다.

AWS CLI

문서 버전 업로드를 시작하려면

다음 initiate-document-upload 예제에서는 새 문서 객체 및 버전 객체를 생성합니다.

```
aws workdocs initiate-document-version-upload \
    --name exampledocname \
    --parent-folder-
id eacd546d952531c633452ed67cac23161aa0d5df2e8061223a59e8f67e7b6189
```

출력:

```
{
    "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-
ddb67663e782e7ce8455ebc962217cf9f9e47b5a9a702e5c84dcccd417da9313",
            "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-
ddb67663e782e7ce8455ebc962217cf9f9e47b5a9a702e5c84dcccd417da9313?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"
        }
    }
}
```

- 자세한 API 내용은 명령 참조InitiateDocumentVersionUpload의 섹션을 참조하세요. AWS CLI

### `remove-all-resource-permissions`

다음 코드 예시에서는 `remove-all-resource-permissions`을 사용하는 방법을 보여 줍니다.

AWS CLI

지정된 리소스에서 모든 권한을 제거하려면

이 예제에서는 지정된 리소스에서 모든 권한을 제거합니다.

명령:

```
aws workdocs remove-all-resource-permissions --resource-
id 1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678
```

출력:

```
None
```

- 자세한 API 내용은 명령 참조RemoveAllResourcePermissions의 섹션을 참조하세요. AWS CLI

### remove-resource-permission

다음 코드 예시에서는 remove-resource-permission을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에서 권한을 제거하려면

이 예제에서는 지정된 보안 주체에 대한 리소스에서 권한을 제거합니다.

명령:

```
aws workdocs remove-resource-permission --resource-
id 1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678 --principal-
id anonymous
```

출력:

```
None
```

- 자세한 API 내용은 명령 참조RemoveResourcePermission의 섹션을 참조하세요. AWS CLI

### update-document-version

다음 코드 예시에서는 update-document-version을 사용하는 방법을 보여 줍니다.

AWS CLI

문서 버전 상태를 활성으로 변경하려면

이 예제에서는 문서 버전의 상태를 활성으로 변경합니다.

명령:

```
aws workdocs update-document-version --document-
id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --version-
id 1521672507741-9f7df0ea5dd0b121c4f3564a0c7c0b4da95cd12c635d3c442af337a88e297920 --
version-status ACTIVE
```

출력:

```
None
```

- 자세한 API 내용은 명령 참조UpdateDocumentVersion의 섹션을 참조하세요. AWS CLI

### update-document

다음 코드 예시에서는 update-document을 사용하는 방법을 보여 줍니다.

AWS CLI

문서를 업데이트하려면

이 예제에서는 문서의 이름과 상위 폴더를 업데이트합니다.

명령:

```
aws workdocs update-document --document-
id 15df51e0335cfcc6a2e4de9dd8be9f22ee40545ad9176f54758dcf903be982d3 --
name updatedDoc --parent-folder-
id 50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08
```

출력:

```
None
```

- 자세한 API 내용은 명령 참조UpdateDocument의 섹션을 참조하세요. AWS CLI

### update-folder

다음 코드 예시에서는 update-folder을 사용하는 방법을 보여 줍니다.

AWS CLI

폴더를 업데이트하려면

이 예제에서는 폴더의 이름과 상위 폴더를 업데이트합니다.

명령:

```
aws workdocs update-folder --folder-
id 50893c0af679524d1a0e0651130ed6d073e1a05f95bd12c42dcde5d35634ed08 --
name exampleFolder1 --parent-folder-
id 1ece93e5fe75315c7407c4967918b4fd9da87ddb2a588e67b7fdaf4a98fde678
```

출력:

```
None
```

- 자세한 API 내용은 명령 참조UpdateFolder의 섹션을 참조하세요. AWS CLI

### update-user

다음 코드 예시에서는 update-user을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자를 업데이트하려면

이 예제에서는 지정된 사용자의 시간대를 업데이트합니다.

명령:

```
aws workdocs update-user --user-
id "S-1-1-11-1111111111-2222222222-3333333333-3333&d-926726012c" --time-zone-
id "America/Los_Angeles"
```

출력:

```
{
  "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"
            }
        }
    }
}
```

- 자세한 API 내용은 명령 참조UpdateUser의 섹션을 참조하세요. AWS CLI

## 를 사용한 Amazon WorkMail 예제 AWS CLI

다음 코드 예제에서는 Amazon 와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 WorkMail.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### **associate-delegate-to-resource**

다음 코드 예시에서는 associate-delegate-to-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 대리인을 추가하려면

다음 associate-delegate-to-resource 명령은 리소스에 대리인을 추가합니다.

```
aws workmail associate-delegate-to-resource \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --resource-id r-68bf2d3b1c0244aab7264c24b9217443 \
```

```
    --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조AssociateDelegateToResource의 섹션을 참조하세요. AWS CLI

### associate-member-to-group

다음 코드 예시에서는 associate-member-to-group을 사용하는 방법을 보여 줍니다.

AWS CLI

그룹에 멤버를 추가하려면

다음 associate-member-to-group 명령은 지정된 멤버를 그룹에 추가합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조AssociateMemberToGroup의 섹션을 참조하세요. AWS CLI

### create-alias

다음 코드 예시에서는 create-alias을 사용하는 방법을 보여 줍니다.

AWS CLI

별칭을 생성하려면

다음 create-alias 명령은 지정된 엔터티(사용자 또는 그룹)에 대한 별칭을 생성합니다.

```
aws workmail create-alias \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --entity-id S-1-1-11-1122222222-2222233333-3333334444-4444 \
    --alias exampleAlias@site.awsapps.com
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조CreateAlias의 섹션을 참조하세요. AWS CLI

### create-group

다음 코드 예시에서는 create-group을 사용하는 방법을 보여 줍니다.

AWS CLI

새 그룹을 생성하려면

다음 create-group 명령은 지정된 조직에 대한 새 그룹을 생성합니다.

```
aws workmail create-group \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --name exampleGroup1
```

출력:

```
{
    "GroupId": "S-1-1-11-1122222222-2222233333-3333334444-4444"
}
```

- 자세한 API 내용은 명령 참조CreateGroup의 섹션을 참조하세요. AWS CLI

### create-resource

다음 코드 예시에서는 create-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

새 리소스를 생성하려면

다음 create-resource 명령은 지정된 조직에 대한 새 리소스(회의실)를 생성합니다.

```
aws workmail create-resource \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --name exampleRoom1 \
    --type ROOM
```

출력:

```
{
    "ResourceId": "r-7afe0efbade843a58cdc10251fce992c"
}
```

- 자세한 API 내용은 명령 참조CreateResource의 섹션을 참조하세요. AWS CLI

### `create-user`

다음 코드 예시에서는 `create-user`을 사용하는 방법을 보여 줍니다.

AWS CLI

새 사용자를 생성하려면

다음 `create-user` 명령은 새 사용자를 생성합니다.

```
aws workmail create-user \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --name exampleName \
    --display-name exampleDisplayName \
    --password examplePa$$w0rd
```

출력:

```
{
    "UserId": "S-1-1-11-1111111111-2222222222-3333333333-3333"
}
```

- 자세한 API 내용은 명령 참조CreateUser의 섹션을 참조하세요. AWS CLI

### `delete-access-control-rule`

다음 코드 예시에서는 `delete-access-control-rule`을 사용하는 방법을 보여 줍니다.

AWS CLI

액세스 제어 규칙을 삭제하려면

다음 `delete-access-control-rule` 예제에서는 지정된 Amazon WorkMail 조직에서 지정된 액세스 제어 규칙을 삭제합니다.

```
aws workmail delete-access-control-rule \
    --organization-id m-n1pq2345678r901st2u3vx45x6789yza \
    --name "myRule"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon WorkMail 관리자 안내서의 액세스 제어 규칙 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteAccessControlRule의 섹션을 참조하세요. AWS CLI

### delete-alias

다음 코드 예시에서는 delete-alias을 사용하는 방법을 보여 줍니다.

AWS CLI

별칭을 삭제하려면

다음 delete-alias 명령은 지정된 엔터티(사용자 또는 그룹)의 별칭을 삭제합니다.

```
aws workmail delete-alias \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --entity-id S-1-1-11-1122222222-2222233333-3333334444-4444 \
    --alias exampleAlias@site.awsapps.com
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeleteAlias의 섹션을 참조하세요. AWS CLI

### delete-group

다음 코드 예시에서는 delete-group을 사용하는 방법을 보여 줍니다.

AWS CLI

기존 그룹을 삭제하려면

다음 delete-group 명령은 Amazon 에서 기존 그룹을 삭제합니다 WorkMail.

```
aws workmail delete-group \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --group-id S-1-1-11-1122222222-2222233333-3333334444-4444
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeleteGroup의 섹션을 참조하세요. AWS CLI

### delete-mailbox-permissions

다음 코드 예시에서는 delete-mailbox-permissions을 사용하는 방법을 보여 줍니다.

AWS CLI

사서함 권한을 삭제하려면

다음 delete-mailbox-permissions 명령은 이전에 사용자 또는 그룹에 부여된 사서함 권한을 삭제합니다. 엔터티는 사서함을 소유한 사용자를 나타내고, 권한 부여자는 권한을 삭제할 사용자 또는 그룹을 나타냅니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeleteMailboxPermissions의 섹션을 참조하세요. AWS CLI

### delete-resource

다음 코드 예시에서는 delete-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

기존 리소스를 삭제하려면

다음 delete-resource 명령은 Amazon 에서 기존 리소스를 삭제합니다 WorkMail.

```
aws workmail delete-resource \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --resource-id r-7afe0efbade843a58cdc10251fce992c
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeleteResource의 섹션을 참조하세요. AWS CLI

### delete-user

다음 코드 예시에서는 delete-user을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 삭제

다음 delete-user 명령은 Amazon WorkMail 및 모든 후속 시스템에서 지정된 사용자를 삭제합니다.

```
aws workmail delete-user \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --user-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeleteUser의 섹션을 참조하세요. AWS CLI

### deregister-from-work-mail

다음 코드 예시에서는 deregister-from-work-mail을 사용하는 방법을 보여 줍니다.

AWS CLI

기존 엔터티를 비활성화하려면

다음 deregister-from-work-mail 명령은 기존 엔터티(사용자, 그룹 또는 리소스)가 Amazon 를 사용하지 못하도록 합니다 WorkMail.

```
aws workmail deregister-from-work-mail \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DeregisterFromWorkMail의 섹션을 참조하세요. AWS CLI

### describe-group

다음 코드 예시에서는 describe-group을 사용하는 방법을 보여 줍니다.

AWS CLI

그룹에 대한 정보를 검색하려면

다음 describe-group 명령은 지정된 그룹에 대한 정보를 검색합니다.

```
aws workmail describe-group \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --group-id S-1-1-11-1122222222-2222233333-3333334444-4444
```

출력:

```
{
    "GroupId": "S-1-1-11-1122222222-2222233333-3333334444-4444",
    "Name": "exampleGroup1",
    "State": "ENABLED"
}
```

- 자세한 API 내용은 명령 참조DescribeGroup의 섹션을 참조하세요. AWS CLI

### describe-organization

다음 코드 예시에서는 describe-organization을 사용하는 방법을 보여 줍니다.

AWS CLI

조직의 정보를 검색하려면

다음 describe-organization 명령은 지정된 Amazon WorkMail 조직에 대한 정보를 검색합니다.

```
aws workmail describe-organization \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27
```

출력:

```
{
    "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"
}
```

자세한 내용은 Amazon WorkMail 관리자 안내서의 조직 작업을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeOrganization의 섹션을 참조하세요. AWS CLI

### describe-resource

다음 코드 예시에서는 describe-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 대한 정보를 검색하려면

다음 describe-resource 명령은 지정된 리소스에 대한 정보를 검색합니다.

```
aws workmail describe-resource \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --resource-id r-7afe0efbade843a58cdc10251fce992c
```

출력:

```
{
    "ResourceId": "r-7afe0efbade843a58cdc10251fce992c",
    "Name": "exampleRoom1",
    "Type": "ROOM",
    "BookingOptions": {
        "AutoAcceptRequests": true,
        "AutoDeclineRecurringRequests": false,
        "AutoDeclineConflictingRequests": true
    },
    "State": "ENABLED"
}
```

- 자세한 API 내용은 명령 참조DescribeResource의 섹션을 참조하세요. AWS CLI

### describe-user

다음 코드 예시에서는 describe-user을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 정보를 검색하려면

다음 describe-user 명령은 지정된 사용자에 대한 정보를 검색합니다.

```
aws workmail describe-user \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --user-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

출력:

```
{
    "UserId": "S-1-1-11-1111111111-2222222222-3333333333-3333",
    "Name": "exampleUser1",
    "Email": "exampleUser1@site.awsapps.com",
    "DisplayName": "",
    "State": "ENABLED",
    "UserRole": "USER",
    "EnabledDate": 1532459261.827
}
```

- 자세한 API 내용은 명령 참조DescribeUser의 섹션을 참조하세요. AWS CLI

### disassociate-delegate-from-resource

다음 코드 예시에서는 disassociate-delegate-from-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에서 멤버를 제거하려면

다음 disassociate-delegate-from-resource 명령은 리소스에서 지정된 멤버를 제거합니다.

```
ws workmail disassociate-delegate-from-resource \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --resource-id r-68bf2d3b1c0244aab7264c24b9217443 \
    --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DisassociateDelegateFromResource의 섹션을 참조하세요. AWS CLI

### disassociate-member-from-group

다음 코드 예시에서는 disassociate-member-from-group을 사용하는 방법을 보여 줍니다.

AWS CLI

그룹에서 멤버를 제거하려면

다음 disassociate-member-from-group 명령은 그룹에서 지정된 멤버를 제거합니다.

```
aws workmail disassociate-member-from-group \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --group-id S-1-1-11-1122222222-2222233333-3333334444-4444 \
    --member-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조DisassociateMemberFromGroup의 섹션을 참조하세요. AWS CLI

### get-access-control-effect

다음 코드 예시에서는 get-access-control-effect을 사용하는 방법을 보여 줍니다.

AWS CLI

액세스 제어 규칙의 효과를 얻으려면

다음 get-access-control-effect 예제에서는 지정된 IP 주소, 액세스 프로토콜 작업 및 사용자 ID에 대한 지정된 Amazon WorkMail 조직의 액세스 제어 규칙의 효과를 검색합니다.

```
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"
```

출력:

```
{
    "Effect": "DENY",
    "MatchedRules": [
        "myRule"
```

```
    ]
}
```

자세한 내용은 Amazon WorkMail 관리자 안내서의 액세스 제어 규칙 작업을 참조하세요.

- 자세한 API 내용은 명령 참조GetAccessControlEffect의 섹션을 참조하세요. AWS CLI

### `get-mailbox-details`

다음 코드 예시에서는 `get-mailbox-details`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자의 사서함 세부 정보를 가져오려면

다음 `get-mailbox-details` 명령은 지정된 사용자의 사서함에 대한 세부 정보를 검색합니다.

```
aws workmail get-mailbox-details \
    --organization-id m-n1pq2345678r901st2u3vx45x6789yza \
    --user-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

출력:

```
{
    "MailboxQuota": 51200,
    "MailboxSize": 0.03890800476074219
}
```

자세한 내용은 Amazon WorkMail 관리자 안내서의 사용자 계정 관리를 참조하세요.

- 자세한 API 내용은 명령 참조GetMailboxDetails의 섹션을 참조하세요. AWS CLI

### `list-access-control-rules`

다음 코드 예시에서는 `list-access-control-rules`을 사용하는 방법을 보여 줍니다.

AWS CLI

액세스 제어 규칙을 나열하려면

다음 `list-access-control-rules` 예제에서는 지정된 Amazon WorkMail 조직에 대한 액세스 제어 규칙을 나열합니다.

```
aws workmail list-access-control-rules \
    --organization-id m-n1pq2345678r901st2u3vx45x6789yza
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 Amazon WorkMail 관리자 안내서의 액세스 제어 규칙 작업을 참조하세요.

- 자세한 API 내용은 명령 참조ListAccessControlRules의 섹션을 참조하세요. AWS CLI

### `list-aliases`

다음 코드 예시에서는 `list-aliases`을 사용하는 방법을 보여 줍니다.

AWS CLI

멤버의 별칭을 나열하려면

다음 `list-aliases` 명령은 지정된 멤버(사용자 또는 그룹)의 별칭을 나열합니다.

```
aws workmail list-aliases \
```

```
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

출력:

```
{
    "Aliases": [
        "exampleAlias@site.awsapps.com",
        "exampleAlias1@site.awsapps.com"
    ]
}
```

- 자세한 API 내용은 명령 참조ListAliases의 섹션을 참조하세요. AWS CLI

### `list-group-members`

다음 코드 예시에서는 `list-group-members`을 사용하는 방법을 보여 줍니다.

AWS CLI

그룹 멤버를 나열하려면

다음 `list-group-members` 명령은 지정된 그룹의 멤버를 나열합니다.

```
aws workmail list-group-members \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --group-id S-1-1-11-1122222222-2222233333-3333334444-4444
```

출력:

```
{
    "Members": [
        {
            "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333",
            "Name": "exampleUser1",
            "Type": "USER",
            "State": "ENABLED",
            "EnabledDate": 1532459261.827
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListGroupMembers의 섹션을 참조하세요. AWS CLI

### list-groups

다음 코드 예시에서는 list-groups을 사용하는 방법을 보여 줍니다.

AWS CLI

그룹 목록을 검색하려면

다음 list-groups 명령은 지정된 조직의 그룹에 대한 요약을 검색합니다.

```
aws workmail list-groups \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListGroups의 섹션을 참조하세요. AWS CLI

### list-mailbox-permissions

다음 코드 예시에서는 list-mailbox-permissions을 사용하는 방법을 보여 줍니다.

AWS CLI

사서함 권한을 검색하려면

다음 list-mailbox-permissions 명령은 지정된 엔터티의 사서함과 연결된 사서함 권한을 검색합니다.

```
aws workmail list-mailbox-permissions \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333
```

출력:

```
{
    "Permissions": [
        {
            "GranteeId": "S-1-1-11-1122222222-2222233333-3333334444-4444",
            "GranteeType": "USER",
            "PermissionValues": [
                "FULL_ACCESS"
            ]
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListMailboxPermissions의 섹션을 참조하세요. AWS CLI

### list-organizations

다음 코드 예시에서는 list-organizations을 사용하는 방법을 보여 줍니다.

AWS CLI

조직 목록을 검색하려면

다음 list-organizations 명령은 삭제되지 않은 조직의 요약을 검색합니다.

```
aws workmail list-organizations
```

출력:

```
{
    "OrganizationSummaries": [
        {
            "OrganizationId": "m-d281d0a2fd824be5b6cd3d3ce909fd27",
```

```
            "Alias": "exampleAlias",
            "State": "Active"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListOrganizations의 섹션을 참조하세요. AWS CLI

### `list-resource-delegates`

다음 코드 예시에서는 `list-resource-delegates`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스의 대리인을 나열하려면

다음 `list-resource-delegates` 명령은 지정된 리소스와 연결된 위임자를 검색합니다.

```
aws workmail list-resource-delegates \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --resource-id r-68bf2d3b1c0244aab7264c24b9217443
```

출력:

```
{
    "Delegates": [
        {
            "Id": "S-1-1-11-1111111111-2222222222-3333333333-3333",
            "Type": "USER"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListResourceDelegates의 섹션을 참조하세요. AWS CLI

### `list-resources`

다음 코드 예시에서는 `list-resources`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스 목록을 검색하려면

다음 `list-resources` 명령은 지정된 조직의 리소스 요약을 검색합니다.

```
aws workmail list-resources \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27
```

출력:

```
{
    "Resources": [
        {
            "Id": "r-7afe0efbade843a58cdc10251fce992c",
            "Name": "exampleRoom1",
            "Type": "ROOM",
            "State": "ENABLED"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListResources의 섹션을 참조하세요. AWS CLI

### `list-tags-for-resource`

다음 코드 예시에서는 `list-tags-for-resource`을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스의 태그를 나열하려면

다음 `list-tags-for-resource` 예제에서는 지정된 Amazon WorkMail 조직의 태그를 나열합니다.

```
aws workmail list-tags-for-resource \
    --resource-arn arn:aws:workmail:us-west-2:111122223333:organization/m-
n1pq2345678r901st2u3vx45x6789yza
```

출력:

```
{
    "Tags": [
        {
            "Key": "priority",
```

```
            "Value": "1"
        }
    ]
}
```

자세한 내용은 Amazon WorkMail 관리자 안내서의 조직 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조ListTagsForResource의 섹션을 참조하세요. AWS CLI

### `list-users`

다음 코드 예시에서는 `list-users`을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자 목록을 검색하려면

다음 `list-users` 명령은 지정된 조직의 사용자 요약을 검색합니다.

```
aws workmail list-users \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27
```

출력:

```
{
    "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"
        }
    ]
}
```

- 자세한 API 내용은 명령 참조ListUsers의 섹션을 참조하세요. AWS CLI

### put-access-control-rule

다음 코드 예시에서는 put-access-control-rule을 사용하는 방법을 보여 줍니다.

AWS CLI

새 액세스 제어 규칙을 배치하려면

다음 put-access-control-rule 예제에서는 지정된 Amazon WorkMail 조직에 대한 지정된 사용자 액세스를 거부합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon WorkMail 관리자 안내서의 액세스 제어 규칙 작업을 참조하세요.

- 자세한 API 내용은 명령 참조PutAccessControlRule의 섹션을 참조하세요. AWS CLI

### put-mailbox-permissions

다음 코드 예시에서는 put-mailbox-permissions을 사용하는 방법을 보여 줍니다.

AWS CLI

사서함 권한을 설정하려면

다음 put-mailbox-permissions 명령은 지정된 권한 부여자(사용자 또는 그룹)에 대한 전체 액세스 권한을 설정합니다. 엔터티는 사서함의 소유자를 나타냅니다.

```
aws workmail put-mailbox-permissions \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --entity-id S-1-1-11-1111111111-2222222222-3333333333-3333 \
    --grantee-id S-1-1-11-1122222222-2222233333-3333334444-4444 \
    --permission-values FULL_ACCESS
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조PutMailboxPermissions의 섹션을 참조하세요. AWS CLI

### register-to-work-mail

다음 코드 예시에서는 register-to-work-mail을 사용하는 방법을 보여 줍니다.

AWS CLI

기존 또는 비활성화된 엔터티를 등록하려면

다음 register-to-work-mail 명령을 사용하면 지정된 기존 엔터티(사용자, 그룹 또는 리소스)가 Amazon 를 사용할 수 있습니다 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
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조RegisterToWorkMail의 섹션을 참조하세요. AWS CLI

### reset-password

다음 코드 예시에서는 reset-password을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자의 암호를 재설정하려면

다음 reset-password 명령은 지정된 사용자의 암호를 재설정합니다.

```
aws workmail reset-password \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --user-id S-1-1-11-1111111111-2222222222-3333333333-3333 \
    --password examplePa$$w0rd
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조ResetPassword의 섹션을 참조하세요. AWS CLI

### tag-resource

다음 코드 예시에서는 tag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스에 태그를 적용하려면

다음 tag-resource 예제에서는 키가 '우선순위'이고 값이 '1'인 태그를 지정된 Amazon WorkMail 조직에 적용합니다.

```
aws workmail tag-resource \
    --resource-arn arn:aws:workmail:us-west-2:111122223333:organization/m-n1pq2345678r901st2u3vx45x6789yza \
    --tags "Key=priority,Value=1"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon WorkMail 관리자 안내서의 조직 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조TagResource의 섹션을 참조하세요. AWS CLI

### untag-resource

다음 코드 예시에서는 untag-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스의 태그를 해제하려면

다음 untag-resource 예제에서는 지정된 Amazon WorkMail 조직에서 지정된 태그를 제거합니다.

```
aws workmail untag-resource \
    --resource-arn arn:aws:workmail:us-west-2:111122223333:organization/m-n1pq2345678r901st2u3vx45x6789yza \
    --tag-keys "priority"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon WorkMail 관리자 안내서의 조직 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조UntagResource의 섹션을 참조하세요. AWS CLI

### update-mailbox-quota

다음 코드 예시에서는 update-mailbox-quota을 사용하는 방법을 보여 줍니다.

AWS CLI

사용자의 사서함 할당량을 업데이트하려면

다음 update-mailbox-quota 명령은 지정된 사용자의 사서함 할당량을 변경합니다.

```
aws workmail update-mailbox-quota \
    --organization-id m-n1pq2345678r901st2u3vx45x6789yza \
    --user-id S-1-1-11-1111111111-2222222222-3333333333-3333 \
    --mailbox-quota 40000
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon WorkMail 관리자 안내서의 사용자 계정 관리를 참조하세요.

- 자세한 API 내용은 명령 참조UpdateMailboxQuota의 섹션을 참조하세요. AWS CLI

### update-primary-email-address

다음 코드 예시에서는 update-primary-email-address을 사용하는 방법을 보여 줍니다.

AWS CLI

기본 이메일 주소를 업데이트하려면

다음 update-primary-email-address 명령은 지정된 엔터티(사용자, 그룹 또는 리소스)의 기본 이메일 주소를 업데이트합니다.

```
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
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조UpdatePrimaryEmailAddress의 섹션을 참조하세요. AWS CLI

**update-resource**

다음 코드 예시에서는 update-resource을 사용하는 방법을 보여 줍니다.

AWS CLI

리소스를 업데이트하려면

다음 update-resource 명령은 지정된 리소스의 이름을 업데이트합니다.

```
aws workmail update-resource \
    --organization-id m-d281d0a2fd824be5b6cd3d3ce909fd27 \
    --resource-id r-7afe0efbade843a58cdc10251fce992c \
    --name exampleRoom2
```

이 명령은 출력을 생성하지 않습니다.

- 자세한 API 내용은 명령 참조UpdateResource의 섹션을 참조하세요. AWS CLI

## 를 사용한 Amazon WorkMail 메시지 흐름 예제 AWS CLI

다음 코드 예제에서는 Amazon WorkMail Message Flow와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

**get-raw-message-content**

다음 코드 예시에서는 get-raw-message-content을 사용하는 방법을 보여 줍니다.

AWS CLI

이메일 메시지의 원시 콘텐츠를 가져오려면

다음 get-raw-message-content 예제에서는 전송 중 이메일 메시지의 원시 콘텐츠를 가져와 라는 텍스트 파일로 보냅니다test.

```
aws workmailmessageflow get-raw-message-content \
    --message-id a1b2cd34-ef5g-6h7j-kl8m-npq9012345rs \
    test
```

명령 실행 test 후 파일 내용:

```
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+--
```

자세한 내용은 Amazon WorkMail 관리자 안내서의 AWS Lambda를 사용하여 메시지 콘텐츠 검색을 참조하세요.

- 자세한 API 내용은 명령 참조GetRawMessageContent의 섹션을 참조하세요. AWS CLI

## WorkSpaces 사용 예제 AWS CLI

다음 코드 예제에서는 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 WorkSpaces.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `create-tags`

다음 코드 예시에서는 `create-tags`을 사용하는 방법을 보여 줍니다.

AWS CLI

에 태그를 추가하려면 WorkSpace

다음 create-tags 예제에서는 지정된 태그를 지정된 에 추가합니다 WorkSpace.

```
aws workspaces create-tags \
    --resource-id ws-dk1xzr417 \
    --tags Key=Department,Value=Finance
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon WorkSpaces 관리 안내서의 WorkSpaces 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조CreateTags의 섹션을 참조하세요. AWS CLI

### create-workspaces

다음 코드 예시에서는 create-workspaces을 사용하는 방법을 보여 줍니다.

AWS CLI

예제 1: 를 생성하려면 AlwaysOn WorkSpace

다음 create-workspaces 예제에서는 지정된 디렉터리와 번들을 사용하여 지정된 사용자에 AlwaysOn WorkSpace 대한 를 생성합니다.

```
aws workspaces create-workspaces \
    --workspaces DirectoryId=d-926722edaf,UserName=Mateo,BundleId=wsb-0zsvgp8fc
```

출력:

```
{
    "FailedRequests": [],
    "PendingRequests": [
        {
            "WorkspaceId": "ws-kcqms853t",
            "DirectoryId": "d-926722edaf",
            "UserName": "Mateo",
            "State": "PENDING",
            "BundleId": "wsb-0zsvgp8fc"
        }
    ]
}
```

예제 2: 생성 AutoStop WorkSpace

다음 create-workspaces 예제에서는 지정된 디렉터리와 번들을 사용하여 지정된 사용자에 AutoStop WorkSpace 대해 를 생성합니다.

```
aws workspaces create-workspaces \
    --
workspaces DirectoryId=d-926722edaf,UserName=Mary,BundleId=wsb-0zsvgp8fc,WorkspaceProperties
```

출력:

```
{
    "FailedRequests": [],
    "PendingRequests": [
        {
            "WorkspaceId": "ws-dk1xzr417",
            "DirectoryId": "d-926722edaf",
            "UserName": "Mary",
            "State": "PENDING",
            "BundleId": "wsb-0zsvgp8fc"
        }
    ]
}
```

예제 3: 사용자 분리 생성 WorkSpace

다음 create-workspaces 예제에서는 사용자 이름을 로 설정하고 WorkSpace 이름[UNDEFINED], 디렉터리 ID 및 번들 ID를 지정 WorkSpace 하여 사용자 분리를 생성합니다.

```
aws workspaces create-workspaces \
    --workspaces
 DirectoryId=d-926722edaf,UserName='"[UNDEFINED]"',WorkspaceName=MaryWorkspace1,BundleId=wsb
```

출력:

```
{
    "FailedRequests": [],
    "PendingRequests": [
        {
            "WorkspaceId": "ws-abcd1234",
            "DirectoryId": "d-926722edaf",
            "UserName": "[UNDEFINED]",
            "State": "PENDING",
            "BundleId": "wsb-0zsvgp8fc",
```

```
            "WorkspaceName": "MaryWorkspace1"
        }
    ]
}
```

자세한 내용은 Amazon WorkSpaces 관리 안내서의 가상 데스크톱 시작을 참조하세요.

- 자세한 API 내용은 명령 참조CreateWorkspaces의 섹션을 참조하세요. AWS CLI

### delete-tags

다음 코드 예시에서는 delete-tags을 사용하는 방법을 보여 줍니다.

AWS CLI

에서 태그를 삭제하려면 WorkSpace

다음 delete-tags 예제에서는 지정된 에서 지정된 태그를 삭제합니다 WorkSpace.

```
aws workspaces delete-tags \
    --resource-id ws-dk1xzr417 \
    --tag-keys Department
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon WorkSpaces 관리 안내서의 WorkSpaces 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조DeleteTags의 섹션을 참조하세요. AWS CLI

### deregister-workspace-directory

다음 코드 예시에서는 deregister-workspace-directory을 사용하는 방법을 보여 줍니다.

AWS CLI

디렉터리 등록을 취소하려면

다음 deregister-workspace-directory 예제에서는 지정된 디렉터리의 등록을 취소합니다.

```
aws workspaces deregister-workspace-directory \
    --directory-id d-926722edaf
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon WorkSpaces 관리 안내서의 에 디렉터리 등록 WorkSpaces을 참조하세요.

- 자세한 API 내용은 명령 참조DeregisterWorkspaceDirectory의 섹션을 참조하세요. AWS CLI

### describe-tags

다음 코드 예시에서는 describe-tags을 사용하는 방법을 보여 줍니다.

AWS CLI

의 태그를 설명하려면 WorkSpace

다음 describe-tags 예제에서는 지정된 에 대한 태그를 설명합니다 WorkSpace.

```
aws workspaces describe-tags \
    --resource-id ws-dk1xzr417
```

출력:

```
{
    "TagList": [
        {
            "Key": "Department",
            "Value": "Finance"
        }
    ]
}
```

자세한 내용은 Amazon WorkSpaces 관리 안내서의 WorkSpaces 리소스 태그 지정을 참조하세요.

- 자세한 API 내용은 명령 참조DescribeTags의 섹션을 참조하세요. AWS CLI

### describe-workspace-bundles

다음 코드 예시에서는 describe-workspace-bundles을 사용하는 방법을 보여 줍니다.

AWS CLI

Amazon에서 제공하는 번들을 나열하려면

다음 describe-workspace-bundles 예제에서는 Amazon에서 제공하는 번들IDs의 이름과 이름을 테이블 형식으로 나열하고 이름으로 정렬합니다.

```
aws workspaces describe-workspace-bundles \
    --owner AMAZON \
    --query "Bundles[*].[Name, BundleId]"
```

출력:

```
[
    [
        "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"
    ],
    ...
]
```

자세한 내용은 Amazon WorkSpaces 관리 안내서의 WorkSpaces 번들 및 이미지를 참조하세요.

- 자세한 API 내용은 명령 참조DescribeWorkspaceBundles의 섹션을 참조하세요. AWS CLI

### describe-workspace-directories

다음 코드 예시에서는 describe-workspace-directories을 사용하는 방법을 보여 줍니다.

AWS CLI

등록된 디렉터리를 설명하려면

다음 describe-workspace-directories 예제에서는 지정된 등록 디렉터리를 설명합니다.

```
aws workspaces describe-workspace-directories \
    --directory-ids d-926722edaf
```

출력:

```
{
    "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"
            }
        }
    ]
}
```

자세한 내용은 Amazon WorkSpaces 관리 안내서의 용 디렉터리 관리를 WorkSpaces 참조하세요.

- 자세한 API 내용은 명령 참조DescribeWorkspaceDirectories의 섹션을 참조하세요. AWS CLI

### `describe-workspaces-connection-status`

다음 코드 예시에서는 `describe-workspaces-connection-status`을 사용하는 방법을 보여 줍니다.

AWS CLI

의 연결 상태를 설명하려면 WorkSpace

다음 `describe-workspaces-connection-status` 예제에서는 지정된 의 연결 상태를 설명합니다 WorkSpace.

```
aws workspaces describe-workspaces-connection-status \
    --workspace-ids ws-dk1xzr417
```

출력:

```
{
    "WorkspacesConnectionStatus": [
        {
            "WorkspaceId": "ws-dk1xzr417",
            "ConnectionState": "CONNECTED",
            "ConnectionStateCheckTimestamp": 1662526214.744
        }
    ]
}
```

자세한 내용은 Amazon WorkSpaces 관리 안내서의 관리를 참조하세요 WorkSpaces.

- 자세한 API 내용은 명령 참조DescribeWorkspacesConnectionStatus의 섹션을 참조하세요. AWS CLI

### `describe-workspaces`

다음 코드 예시에서는 `describe-workspaces`을 사용하는 방법을 보여 줍니다.

AWS CLI

를 설명하려면 WorkSpace

다음 `describe-workspaces` 예제에서는 지정된 를 설명합니다 WorkSpace.

```
aws workspaces describe-workspaces \
    --workspace-ids ws-dk1xzr417
```

출력:

```
{
    "Workspaces": [
        {
            "WorkspaceId": "ws-dk1xzr417",
            "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": []
        }
    ]
}
```

자세한 내용은 Amazon WorkSpaces 관리 안내서의 관리를 참조하세요 WorkSpaces.

- 자세한 API 내용은 명령 참조DescribeWorkspaces의 섹션을 참조하세요. AWS CLI

### `migrate-workspace`

다음 코드 예시에서는 `migrate-workspace`을 사용하는 방법을 보여 줍니다.

AWS CLI

를 마이그레이션하려면 WorkSpace

다음 `migrate-workspace` 예제에서는 지정된 를 지정된 번들 WorkSpace 로 마이그레이션합니다.

```
aws workspaces migrate-workspace \
    --source-workspace-id ws-dk1xzr417 \
    --bundle-id wsb-j4dky1gs4
```

출력:

```
{
    "SourceWorkspaceId": "ws-dk1xzr417",
    "TargetWorkspaceId": "ws-x5h1lbkp5"
}
```

자세한 내용은 Amazon WorkSpaces 관리 안내서의 마이그레이션 WorkSpace을 참조하세요.

- 자세한 API 내용은 명령 참조MigrateWorkspace의 섹션을 참조하세요. AWS CLI

### `modify-workspace-creation-properties`

다음 코드 예시에서는 `modify-workspace-creation-properties`을 사용하는 방법을 보여 줍니다.

AWS CLI

디렉터리의 WorkSpace 생성 속성을 수정하려면

다음 `modify-workspace-creation-properties` 예제에서는 지정된 디렉터리에 대한 `EnableInternetAccess` 속성을 활성화합니다. 이렇게 하면 디렉터리에 대해 WorkSpaces 생성된 에 대한 퍼블릭 IP 주소가 자동으로 할당됩니다.

```
aws workspaces modify-workspace-creation-properties \
```

```
    --resource-id d-926722edaf \
    --workspace-creation-properties EnableInternetAccess=true
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon WorkSpaces 관리 안내서의 에 대한 디렉터리 세부 정보 업데이트를 WorkSpaces 참조하세요.

- 자세한 API 내용은 명령 참조ModifyWorkspaceCreationProperties의 섹션을 참조하세요. AWS CLI

### modify-workspace-properties

다음 코드 예시에서는 modify-workspace-properties을 사용하는 방법을 보여 줍니다.

AWS CLI

의 실행 모드를 수정하려면 WorkSpace

다음 modify-workspace-properties 예제에서는 지정된 의 실행 모드를 WorkSpace 로 설정합니다AUTO_STOP.

```
aws workspaces modify-workspace-properties \
    --workspace-id ws-dk1xzr417 \
    --workspace-properties RunningMode=AUTO_STOP
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon WorkSpaces 관리 안내서의 수정 WorkSpace을 참조하세요.

- 자세한 API 내용은 명령 참조ModifyWorkspaceProperties의 섹션을 참조하세요. AWS CLI

### modify-workspace-state

다음 코드 예시에서는 modify-workspace-state을 사용하는 방법을 보여 줍니다.

AWS CLI

의 상태를 수정하려면 WorkSpace

다음 modify-workspace-state 예제에서는 지정된 의 상태를 WorkSpace 로 설정합니다ADMIN_MAINTENANCE.

```
aws workspaces modify-workspace-state \
    --workspace-id ws-dk1xzr417 \
    --workspace-state ADMIN_MAINTENANCE
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon WorkSpaces 관리 안내서의 WorkSpace 유지 관리를 참조하세요.

- 자세한 API 내용은 명령 참조ModifyWorkspaceState의 섹션을 참조하세요. AWS CLI

### reboot-workspaces

다음 코드 예시에서는 reboot-workspaces을 사용하는 방법을 보여 줍니다.

AWS CLI

재부팅하려면 WorkSpace

다음 reboot-workspaces 예제에서는 지정된 를 재부팅합니다 WorkSpace.

```
aws workspaces reboot-workspaces \
    --reboot-workspace-requests ws-dk1xzr417
```

출력:

```
{
    "FailedRequests": []
}
```

자세한 내용은 Amazon WorkSpaces 관리 안내서의 재부팅 WorkSpace을 참조하세요.

- 자세한 API 내용은 명령 참조RebootWorkspaces의 섹션을 참조하세요. AWS CLI

### rebuild-workspaces

다음 코드 예시에서는 rebuild-workspaces을 사용하는 방법을 보여 줍니다.

AWS CLI

를 다시 빌드하려면 WorkSpace

다음 rebuild-workspaces 예제에서는 지정된 를 재구축합니다 WorkSpace.

```
aws workspaces rebuild-workspaces \
    --rebuild-workspace-requests ws-dk1xzr417
```

출력:

```
{
    "FailedRequests": []
}
```

자세한 내용은 Amazon WorkSpaces 관리 안내서의 재구축 WorkSpace을 참조하세요.

- 자세한 API 내용은 명령 참조RebuildWorkspaces의 섹션을 참조하세요. AWS CLI

### register-workspace-directory

다음 코드 예시에서는 register-workspace-directory을 사용하는 방법을 보여 줍니다.

AWS CLI

디렉터리를 등록하려면

다음 register-workspace-directory 예제에서는 Amazon 에 사용할 지정된 디렉터리를 등록합니다 WorkSpaces.

```
aws workspaces register-workspace-directory \
    --directory-id d-926722edaf \
    --no-enable-work-docs
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon WorkSpaces 관리 안내서의 에 디렉터리 등록 WorkSpaces을 참조하세요.

- 자세한 API 내용은 명령 참조RegisterWorkspaceDirectory의 섹션을 참조하세요. AWS CLI

### restore-workspace

다음 코드 예시에서는 restore-workspace을 사용하는 방법을 보여 줍니다.

AWS CLI

를 복원하려면 WorkSpace

다음 `restore-workspace` 예제에서는 지정된 를 복원합니다 WorkSpace.

```
aws workspaces restore-workspace \
    --workspace-id ws-dk1xzr417
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 Amazon WorkSpaces 관리 안내서의 복원 WorkSpace을 참조하세요.

- 자세한 API 내용은 명령 참조RestoreWorkspace의 섹션을 참조하세요. AWS CLI

### `start-workspaces`

다음 코드 예시에서는 `start-workspaces`을 사용하는 방법을 보여 줍니다.

AWS CLI

를 시작하려면 AutoStop WorkSpace

다음 `start-workspaces` 예제에서는 지정된 를 시작합니다 WorkSpace. 의 실행 모드는 여야 WorkSpace 합니다AutoStop.

```
aws workspaces start-workspaces \
    --start-workspace-requests WorkspaceId=ws-dk1xzr417
```

출력:

```
{
    "FailedRequests": []
}
```

자세한 내용은 Amazon WorkSpaces 관리 안내서의 중지 및 시작 AutoStop WorkSpace을 참조하세요.

- 자세한 API 내용은 명령 참조StartWorkspaces의 섹션을 참조하세요. AWS CLI

### `stop-workspaces`

다음 코드 예시에서는 `stop-workspaces`을 사용하는 방법을 보여 줍니다.

AWS CLI

를 중지하려면 AutoStop WorkSpace

다음 `stop-workspaces` 예제에서는 지정된 를 중지합니다 WorkSpace. 의 실행 모드는 여야 WorkSpace 합니다`AutoStop`.

```
aws workspaces stop-workspaces \
    --stop-workspace-requests WorkspaceId=ws-dk1xzr417
```

출력:

```
{
    "FailedRequests": []
}
```

자세한 내용은 Amazon WorkSpaces 관리 안내서의 중지 및 시작 AutoStop WorkSpace을 참조하세요.

- 자세한 API 내용은 명령 참조StopWorkspaces의 섹션을 참조하세요. AWS CLI

### `terminate-workspaces`

다음 코드 예시에서는 `terminate-workspaces`을 사용하는 방법을 보여 줍니다.

AWS CLI

를 종료하려면 WorkSpace

다음 `terminate-workspaces` 예제에서는 지정된 작업 영역을 종료합니다.

```
aws workspaces terminate-workspaces \
    --terminate-workspace-requests ws-dk1xzr417
```

출력:

```
{
    "FailedRequests": []
}
```

자세한 내용은 Amazon WorkSpaces 관리 안내서의 삭제 WorkSpace를 참조하세요.

- 자세한 API 내용은 명령 참조TerminateWorkspaces의 섹션을 참조하세요. AWS CLI

## 를 사용한 X-Ray 예제 AWS CLI

다음 코드 예제에서는 X-Ray와 AWS Command Line Interface 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `batch-traces-get`

다음 코드 예시에서는 `batch-traces-get`을 사용하는 방법을 보여 줍니다.

AWS CLI

추적 목록을 가져오려면

다음 `batch-get-traces` 예제에서는 ID로 지정된 트레이스 목록을 검색합니다. 전체 트레이스에는 동일한 트레이스 ID로 수신된 모든 세그먼트 문서로부터 컴파일된 각 세그먼트의 문서가 포함됩니다.

```
aws xray batch-get-traces \
    --trace-ids 1-5d82881a-0a9126e92a73e971eed891b9
```

출력:

```
{
    "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\":\"TPEIDNDUROMLPOV17U4A79555NVV4KQNSO5AEMVJF66Q9ASUAAJG
\",\"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\":
\"TPEIDNDUROMLPOV17U4A79555NVV4KQNSO5AEMVJF66Q9ASUAAJG\",\"resource_names\":
[\"scorekeep-user\"]},\"trace_id\":\"1-5d82881a-0a9126e92a73e971eed891b9\",\"origin
\":\"AWS::DynamoDB::Table\"}"
                }
            ]
        }
    ],
    "UnprocessedTraceIds": []
}
```


자세한 내용은 AWS X-Ray 개발자 안내서의 API 에서 AWS CLI X-Ray 사용을 참조하세요. AWS

- 자세한 API 내용은 명령 참조BatchTracesGet의 섹션을 참조하세요. AWS CLI

### create-group

다음 코드 예시에서는 create-group을 사용하는 방법을 보여 줍니다.

AWS CLI

그룹을 생성하려면

다음 create-group 예제에서는 라는 그룹 리소스를 생성합니다AdminGroup. 그룹은 그룹의 기준을 오류 또는 오류를 유발하는 특정 서비스와 관련된 세그먼트로 정의하는 필터 표현식을 가져옵니다.

```
aws xray create-group \
   --group-name "AdminGroup" \
   --filter-expression "service(\"mydomain.com\") {fault OR error}"
```

출력:

```
{
    "GroupName": "AdminGroup",
    "GroupARN": "arn:aws:xray:us-west-2:123456789012:group/AdminGroup/123456789",
    "FilterExpression": "service(\"mydomain.com\") {fault OR error}"
}
```

자세한 내용은 AWS X-Ray 개발자 안내서의 X-Ray를 사용한 샘플링, 그룹 및 암호화 설정 구성을 참조하세요API. AWS

- 자세한 API 내용은 명령 참조CreateGroup의 섹션을 참조하세요. AWS CLI

### create-sampling-rule

다음 코드 예시에서는 create-sampling-rule을 사용하는 방법을 보여 줍니다.

AWS CLI

샘플링 규칙을 생성하려면

다음 create-sampling-rule 예제에서는 계측된 애플리케이션의 샘플링 동작을 제어하는 규칙을 생성합니다. 규칙은 JSON 파일에서 제공됩니다. 대부분의 샘플링 규칙 필드는 규칙을 생성하는 데 필요합니다.

```
aws xray create-sampling-rule \
```

```
    --cli-input-json file://9000-base-scorekeep.json
```

9000-base-scorekeep.json의 콘텐츠:

```
{
    "SamplingRule": {
        "RuleName": "base-scorekeep",
        "ResourceARN": "*",
        "Priority": 9000,
        "FixedRate": 0.1,
        "ReservoirSize": 5,
        "ServiceName": "Scorekeep",
        "ServiceType": "*",
        "Host": "*",
        "HTTPMethod": "*",
        "URLPath": "*",
        "Version": 1
    }
}
```

출력:

```
{
    "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
    }
```

```
}
```

자세한 내용은 AWS X-Ray 개발자 안내서의 X-Ray를 사용한 샘플링, 그룹 및 암호화 설정 구성을 참조하세요API. AWS

- 자세한 API 내용은 명령 참조CreateSamplingRule의 섹션을 참조하세요. AWS CLI

### delete-group

다음 코드 예시에서는 delete-group을 사용하는 방법을 보여 줍니다.

AWS CLI

그룹을 삭제하려면

다음 delete-group 예제에서는 지정된 그룹 리소스를 삭제합니다.

```
aws xray delete-group \
    --group-name "AdminGroup" \
    --group-arn "arn:aws:xray:us-east-2:123456789012:group/AdminGroup/123456789"
```

이 명령은 출력을 생성하지 않습니다.

자세한 내용은 AWS X-Ray 개발자 안내서의 X-Ray를 사용한 샘플링, 그룹 및 암호화 설정 구성을 참조하세요API. AWS

- 자세한 API 내용은 명령 참조DeleteGroup의 섹션을 참조하세요. AWS CLI

### delete-sampling-rule

다음 코드 예시에서는 delete-sampling-rule을 사용하는 방법을 보여 줍니다.

AWS CLI

샘플링 규칙을 삭제하려면

다음 delete-sampling-rule 예제에서는 지정된 샘플링 규칙을 삭제합니다. 그룹 이름 또는 그룹 를 사용하여 그룹을 지정할 수 있습니다ARN.

```
aws xray delete-sampling-rule \
    --rule-name polling-scorekeep
```

출력:

```
{
    "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
    }
}
```

자세한 내용은 AWS X-Ray 개발자 안내서의 X-Ray를 사용한 샘플링, 그룹 및 암호화 설정 구성을 참조하세요API. AWS

- 자세한 API 내용은 명령 참조DeleteSamplingRule의 섹션을 참조하세요. AWS CLI

### `get-encryption-config`

다음 코드 예시에서는 `get-encryption-config`을 사용하는 방법을 보여 줍니다.

AWS CLI

암호화 구성을 검색하려면

다음 `get-encryption-config` 예제에서는 AWS X-Ray 데이터에 대한 현재 암호화 구성을 검색합니다.

```
aws xray get-encryption-config
```

출력:

```
{
    "EncryptionConfig": {
        "KeyId": "ae4aa6d49-a4d8-9df9-a475-4ff6d7898456",
        "Status": "ACTIVE",
        "Type": "NONE"
    }
}
```

자세한 내용은 AWS X-Ray 개발자 안내서의 X-Ray를 사용한 샘플링, 그룹 및 암호화 설정 구성을 참조하세요API. AWS

- 자세한 API 내용은 명령 참조GetEncryptionConfig의 섹션을 참조하세요. AWS CLI

### `get-group`

다음 코드 예시에서는 `get-group`을 사용하는 방법을 보여 줍니다.

AWS CLI

그룹을 검색하려면

다음 `get-group` 예제에서는 지정된 그룹 리소스에 대한 세부 정보를 표시합니다. 세부 정보에는 그룹 이름, 그룹 ARN및 해당 그룹의 기준을 정의하는 필터 표현식이 포함됩니다. 그룹은 에서 검색할 수도 있습니다ARN.

```
aws xray get-group \
    --group-name "AdminGroup"
```

출력:

```
{
    "Group": [
        {
            "GroupName": "AdminGroup",
            "GroupARN": "arn:aws:xray:us-west-2:123456789012:group/
AdminGroup/123456789",
            "FilterExpression": "service(\"mydomain.com\") {fault OR error}"
        }
    ]
```

```
}
```

자세한 내용은 AWS X-Ray 개발자 안내서의 X-Ray를 사용한 샘플링, 그룹 및 암호화 설정 구성을 참조하세요API. AWS

- 자세한 API 내용은 명령 참조GetGroup의 섹션을 참조하세요. AWS CLI

### `get-groups`

다음 코드 예시에서는 `get-groups`을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 그룹을 검색하려면

다음 예제에서는 모든 활성 그룹에 대한 세부 정보를 표시합니다.

```
aws xray get-groups
```

출력:

```
{
    "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"
        }
    ]
}
```

자세한 내용은 AWS X-Ray 개발자 안내서의 X-Ray를 사용한 샘플링, 그룹 및 암호화 설정 구성을 참조하세요API. AWS

- 자세한 API 내용은 명령 참조GetGroups의 섹션을 참조하세요. AWS CLI

### get-sampling-rules

다음 코드 예시에서는 get-sampling-rules을 사용하는 방법을 보여 줍니다.

AWS CLI

모든 샘플링 규칙을 검색하려면

다음 get-sampling-rules 예제에서는 사용 가능한 모든 샘플링 규칙에 대한 세부 정보를 표시합니다.

```
aws xray get-sampling-rules
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 X-Ray 개발자 안내서의 X-Ray에서 샘플링 규칙 사용을 API 참조하세요. AWS

- 자세한 API 내용은 명령 참조GetSamplingRules의 섹션을 참조하세요. AWS CLI

### `get-sampling-targets`

다음 코드 예시에서는 `get-sampling-targets`을 사용하는 방법을 보여 줍니다.

AWS CLI

샘플링 할당량을 요청하려면

다음 `get-sampling-targets` 예제에서는 서비스가 요청을 샘플링하는 데 사용하는 규칙에 대한 샘플링 할당량을 요청합니다. AWS X-Ray의 응답에는 저장소에서 빌리는 대신 사용할 수 있는 할당량이 포함됩니다.

```
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 } ]'
```

출력:

```
{
    "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": []
}
```

자세한 내용은 X-Ray 개발자 안내서의 X-Ray에서 샘플링 규칙 사용을 API 참조하세요. AWS

- 자세한 API 내용은 명령 참조GetSamplingTargets의 섹션을 참조하세요. AWS CLI

### get-service-graph

다음 코드 예시에서는 get-service-graph을 사용하는 방법을 보여 줍니다.

AWS CLI

서비스 그래프를 가져오려면

다음 예제에서는 지정된 기간 내에 수신 요청을 처리하는 서비스와 그 결과로 호출하는 다운스트림 서비스를 설명하는 문서를 표시합니다.

```
aws xray get-service-graph \
    --start-time 1568835392.0
    --end-time 1568835446.0
```

출력:

```
{
    "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
}
```

자세한 내용은 AWS X-Ray 개발자 안내서의 API AWS CLI에서 X-Ray 사용을 참조하세요. AWS

- 자세한 API 내용은 명령 참조GetServiceGraph의 섹션을 참조하세요. AWS CLI

### `get-trace-summaries`

다음 코드 예시에서는 `get-trace-summaries`을 사용하는 방법을 보여 줍니다.

AWS CLI

추적 요약을 가져오려면

다음 `get-trace-summaries` 예제에서는 지정된 기간 내에 사용 가능한 추적에 대한 IDs 및 메타데이터를 검색합니다.

```
aws xray get-trace-summaries \
    --start-time 1568835392.0 \
    --end-time 1568835446.0
```

출력:

```
[
    "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"
]
```

자세한 내용은 AWS X-Ray 개발자 안내서의 API AWS CLI 에서 X-Ray 사용을 참조하세요. AWS

- 자세한 API 내용은 명령 참조GetTraceSummaries의 섹션을 참조하세요. AWS CLI

### put-encryption-config

다음 코드 예시에서는 put-encryption-config을 사용하는 방법을 보여 줍니다.

AWS CLI

암호화 구성을 업데이트하려면

다음 put-encryption-config``example updates the encryption configuration for AWS X-Ray data to use the default AWS managed KMS key ``aws/xray.

```
aws xray put-encryption-config \
    --type KMS \
    --key-id alias/aws/xray
```

출력:

```
{
    "EncryptionConfig": {
        "KeyId": "arn:aws:kms:us-west-2:123456789012:key/c234g4e8-39e9-4gb0-84e2-
b0ea215cbba5",
        "Status": "UPDATING",
        "Type": "KMS"
    }
}
```

자세한 내용은 AWS X-Ray 개발자 안내서의 X-Ray를 사용한 샘플링, 그룹 및 암호화 설정 구성을 참조하세요API. AWS

- 자세한 API 내용은 명령 참조PutEncryptionConfig의 섹션을 참조하세요. AWS CLI

### put-trace-segments

다음 코드 예시에서는 put-trace-segments을 사용하는 방법을 보여 줍니다.

AWS CLI

세그먼트를 업로드하려면

다음 put-trace-segments 예제에서는 세그먼트 문서를 AWS X-Ray에 업로드합니다. 세그먼트 문서는 JSON 세그먼트 문서 목록으로 사용됩니다.

```
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\":
\"SCAU23OM6M8FO38UASGC7785ARVV4KQNSO5AEMVJF66Q9ASUAAJG\",\"resource_names\":
[\"scorekeep-session-xray\"]},\"origin\":\"AWS::DynamoDB::Table\"}"
```

출력:

```
{
    "UnprocessedTraceSegments": []
}
```

자세한 내용은 AWS X-Ray 개발자 안내서의 X-Ray로 추적 데이터 전송을 참조하세요. AWS

- 자세한 API 내용은 명령 참조PutTraceSegments의 섹션을 참조하세요. AWS CLI

### update-group

다음 코드 예시에서는 update-group을 사용하는 방법을 보여 줍니다.

AWS CLI

그룹을 업데이트하려면

다음 update-group 예제에서는 라는 그룹으로 트레이스를 수락할 기준을 업데이트합니다AdminGroup. 그룹 이름 또는 그룹 를 사용하여 원하는 그룹을 지정할 수 있습니다ARN.

```
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}"
```

출력:

```
{
    "GroupName": "AdminGroup",
    "GroupARN": "arn:aws:xray:us-east-2:123456789012:group/AdminGroup/123456789",
    "FilterExpression": "service(\"mydomain.com\") {fault}"
}
```

자세한 내용은 AWS X-Ray 개발자 안내서의 X-Ray를 사용한 샘플링, 그룹 및 암호화 설정 구성을 참조하세요API. AWS

- 자세한 API 내용은 명령 참조UpdateGroup의 섹션을 참조하세요. AWS CLI

### update-sampling-rule

다음 코드 예시에서는 update-sampling-rule을 사용하는 방법을 보여 줍니다.

AWS CLI

샘플링 규칙을 업데이트하려면

다음 update-sampling-rule 예제에서는 샘플링 규칙의 구성을 수정합니다. 규칙은 JSON 파일에서 사용됩니다. 업데이트 중인 필드만 필요합니다.

```
aws xray update-sampling-rule \
   --cli-input-json file://1000-default.json
```

1000-default.json의 콘텐츠:

```
{
    "SamplingRuleUpdate": {
        "RuleName": "Default",
        "FixedRate": 0.01,
        "ReservoirSize": 0
    }
}
```

출력:

```
{
    "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
        }
    ]
}
```

자세한 내용은 AWS X-Ray 개발자 안내서의 X-Ray를 사용한 샘플링, 그룹 및 암호화 설정 구성을 참조하세요API. AWS

- 자세한 API 내용은 명령 참조UpdateSamplingRule의 섹션을 참조하세요. AWS CLI

# AWS CLI Bash 스크립트 코드 예제 사용

이 주제의 코드 예제에서는 와 AWS Command Line Interface 함께 Bash 스크립트를 사용하는 방법을 보여줍니다 AWS.

기본 사항은 서비스 내에서 필수 작업을 수행하는 방법을 보여주는 코드 예제입니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

시나리오는 동일한 서비스 내에서 또는 다른 AWS 서비스와 결합된 상태에서 여러 함수를 호출하여 특정 태스크를 수행하는 방법을 보여주는 코드 예제입니다.

서비스

- Bash 스크립트 AWS CLI 와 함께 를 사용하는 DynamoDB 예제
- 를 Bash 스크립트 AWS CLI 와 함께 사용하는 Amazon EC2 예제
- HealthImaging 를 Bash 스크립트 AWS CLI 와 함께 사용하는 예제
- IAM 를 Bash 스크립트 AWS CLI 와 함께 사용하는 예제
- Bash 스크립트 AWS CLI 와 함께 를 사용하는 Amazon S3 예제
- AWS STS 를 Bash 스크립트 AWS CLI 와 함께 사용하는 예제

# Bash 스크립트 AWS CLI 와 함께 를 사용하는 DynamoDB 예제

다음 코드 예제에서는 DynamoDB 와 AWS Command Line Interface 함께 Bash 스크립트를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

기본 사항은 서비스 내에서 필수 작업을 수행하는 방법을 보여주는 코드 예제입니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 기본 사항

기본 사항 알아보기

다음 코드 예시는 다음과 같은 작업을 수행하는 방법을 보여줍니다.

- 영화 데이터를 저장할 수 있는 테이블을 생성합니다.
- 테이블에 하나의 영화를 추가하고 가져오고 업데이트합니다.
- 샘플 JSON 파일에서 테이블에 영화 데이터를 씁니다.
- 특정 연도에 개봉된 영화를 쿼리합니다.
- 특정 연도 범위 동안 개봉된 영화를 스캔합니다.
- 테이블에서 영화를 삭제한 다음, 테이블을 삭제합니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

DynamoDB 시작 시나리오입니다.

```
###############################################################################
# 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
}
```

이 시나리오에 사용된 DynamoDB 함수입니다.

```
###############################################################################
# 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 getopts command in a function.

  ######################################
  # Function usage explanation
  ######################################
  function usage() {
    echo "function dynamodb_create_table"
    echo "Creates an Amazon DynamoDB table."
    echo " -n table_name  -- The name of the table to create."
    echo " -a attribute_definitions -- JSON file path of a list of attributes and
 their types."
    echo " -k key_schema -- JSON file path of a list of attributes and their key
 types."
```

```
    echo " -p provisioned_throughput -- Provisioned throughput settings for the
table."
    echo ""
 }

 # Retrieve the calling parameters.
 while getopts "n:a:k:p:h" option; do
   case "${option}" in
     n) table_name="${OPTARG}" ;;
     a) attribute_definitions="${OPTARG}" ;;
     k) key_schema="${OPTARG}" ;;
     p) provisioned_throughput="${OPTARG}" ;;
     h)
       usage
       return 0
       ;;
     \?)
       echo "Invalid parameter"
       usage
       return 1
       ;;
   esac
 done
 export OPTIND=1

 if [[ -z "$table_name" ]]; then
   errecho "ERROR: You must provide a table name with the -n parameter."
   usage
   return 1
 fi

 if [[ -z "$attribute_definitions" ]]; then
   errecho "ERROR: You must provide an attribute definitions json file path the -a
parameter."
   usage
   return 1
 fi

 if [[ -z "$key_schema" ]]; then
   errecho "ERROR: You must provide a key schema json file path the -k parameter."
   usage
   return 1
 fi
```

```
  if [[ -z "$provisioned_throughput" ]]; then
    errecho "ERROR: You must provide a provisioned throughput json file path the -p
 parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    table_name:   $table_name"
  iecho "    attribute_definitions:   $attribute_definitions"
  iecho "    key_schema:   $key_schema"
  iecho "    provisioned_throughput:   $provisioned_throughput"
  iecho ""

  response=$(aws dynamodb create-table \
    --table-name "$table_name" \
    --attribute-definitions file://"$attribute_definitions" \
    --key-schema file://"$key_schema" \
    --provisioned-throughput "$provisioned_throughput")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-table operation failed.$response"
    return 1
  fi

  return 0
}

###############################################################################
# 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 getopts 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
}

###############################################################################
# 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 getopts 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 getopts 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 getopts 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 getopts "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 getopts 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 getopts 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 getopts 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
}

###############################################################################
# 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 getopts 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 getopts 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 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

  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
}
```

이 시나리오에 사용된 유틸리티 함수입니다.

```
###############################################################################
# function iecho
#
# This function enables the script to display the specified text only if
# the global variable $VERBOSE is set to true.
###############################################################################
function iecho() {
  if [[ $VERBOSE == true ]]; then
    echo "$@"
  fi
}

###############################################################################
```

```
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
#############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

#############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# See https://docs.aws.amazon.com/cli/latest/topic/return-codes.html#cli-aws-help-
return-codes.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
#############################################################################
function aws_cli_error_log() {
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- API 자세한 내용은 AWS CLI 명령 참조의 다음 주제를 참조하세요.
  - BatchWriteItem
  - CreateTable
  - DeleteItem
  - DeleteTable
  - DescribeTable
  - GetItem
  - PutItem
  - Query
  - Scan
  - UpdateItem

## 작업

### `BatchGetItem`

다음 코드 예시에서는 `BatchGetItem`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
##############################################################################
# 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 getopts 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 getopts "i:h" option; do
    case "${option}" in
      i) item="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$item" ]]; then
    errecho "ERROR: You must provide an item with the -i parameter."
    usage
    return 1
  fi

  response=$(aws dynamodb batch-get-item \
    --request-items file://"$item")
  local error_code=${?}
```

```
  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports batch-get-item operation failed.$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

이 예제에 사용된 유틸리티 함수

```
#############################################################################
# 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
}
```

- 자세한 API 내용은 명령 참조BatchGetItem의 섹션을 참조하세요. AWS CLI

### `BatchWriteItem`

다음 코드 예시에서는 `BatchWriteItem`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
##############################################################################
# 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 getopts 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 getopts "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
}
```

이 예제에 사용된 유틸리티 함수

```
############################################################################
# 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
}
```

- 자세한 API 내용은 명령 참조BatchWriteItem의 섹션을 참조하세요. AWS CLI

### `CreateTable`

다음 코드 예시에서는 `CreateTable`을 사용하는 방법을 보여 줍니다.

## AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
##############################################################################
# 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 getopts command in a function.

  ######################################
  # Function usage explanation
  ######################################
  function usage() {
    echo "function dynamodb_create_table"
    echo "Creates an Amazon DynamoDB table."
    echo " -n table_name  -- The name of the table to create."
    echo " -a attribute_definitions -- JSON file path of a list of attributes and
 their types."
    echo " -k key_schema -- JSON file path of a list of attributes and their key
 types."
    echo " -p provisioned_throughput -- Provisioned throughput settings for the
 table."
    echo ""
```

```
 }

 # Retrieve the calling parameters.
 while getopts "n:a:k:p:h" option; do
   case "${option}" in
     n) table_name="${OPTARG}" ;;
     a) attribute_definitions="${OPTARG}" ;;
     k) key_schema="${OPTARG}" ;;
     p) provisioned_throughput="${OPTARG}" ;;
     h)
       usage
       return 0
       ;;
     \?)
       echo "Invalid parameter"
       usage
       return 1
       ;;
   esac
 done
 export OPTIND=1

 if [[ -z "$table_name" ]]; then
   errecho "ERROR: You must provide a table name with the -n parameter."
   usage
   return 1
 fi

 if [[ -z "$attribute_definitions" ]]; then
   errecho "ERROR: You must provide an attribute definitions json file path the -a
parameter."
   usage
   return 1
 fi

 if [[ -z "$key_schema" ]]; then
   errecho "ERROR: You must provide a key schema json file path the -k parameter."
   usage
   return 1
 fi

 if [[ -z "$provisioned_throughput" ]]; then
   errecho "ERROR: You must provide a provisioned throughput json file path the -p
parameter."
```

```
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    table_name:   $table_name"
  iecho "    attribute_definitions:   $attribute_definitions"
  iecho "    key_schema:   $key_schema"
  iecho "    provisioned_throughput:   $provisioned_throughput"
  iecho ""

  response=$(aws dynamodb create-table \
    --table-name "$table_name" \
    --attribute-definitions file://"$attribute_definitions" \
    --key-schema file://"$key_schema" \
    --provisioned-throughput "$provisioned_throughput")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-table operation failed.$response"
    return 1
  fi

  return 0
}
```

이 예제에 사용된 유틸리티 함수

```
###############################################################################
# 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
```

```
}
```

- 자세한 API 내용은 명령 참조CreateTable의 섹션을 참조하세요. AWS CLI

### `DeleteItem`

다음 코드 예시에서는 `DeleteItem`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
###############################################################################
# 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 getopts 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

}
```

이 예제에 사용된 유틸리티 함수

```
###############################################################################
# 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
}
```

- 자세한 API 내용은 명령 참조DeleteItem의 섹션을 참조하세요. AWS CLI

### `DeleteTable`

다음 코드 예시에서는 `DeleteTable`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
###############################################################################
# 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 getopts 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 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

  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
}
```

이 예제에 사용된 유틸리티 함수

```
###############################################################################
# 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
}
```

- 자세한 API 내용은 명령 참조DeleteTable의 섹션을 참조하세요. AWS CLI

### `DescribeTable`

다음 코드 예시에서는 `DescribeTable`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
###############################################################################
# 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 getopts 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
}
```

이 예제에 사용된 유틸리티 함수

```
##############################################################################
# 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
}
```

- 자세한 API 내용은 명령 참조DescribeTable의 섹션을 참조하세요. AWS CLI

### `GetItem`

다음 코드 예시에서는 `GetItem`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
###############################################################################
# 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 getopts 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
}
```

이 예제에 사용된 유틸리티 함수

```
###############################################################################
# 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
}
```

- 자세한 API 내용은 명령 참조GetItem의 섹션을 참조하세요. AWS CLI

### `ListTables`

다음 코드 예시에서는 `ListTables`을 사용하는 방법을 보여 줍니다.

## AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
##############################################################################
# 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
}
```

이 예제에 사용된 유틸리티 함수

```
##############################################################################
# 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
}
```

- 자세한 API 내용은 명령 참조ListTables의 섹션을 참조하세요. AWS CLI

### `PutItem`

다음 코드 예시에서는 `PutItem`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
###############################################################################
# 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 getopts 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

}
```

이 예제에 사용된 유틸리티 함수

```
##############################################################################
# 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
}
```

- 자세한 API 내용은 명령 참조PutItem의 섹션을 참조하세요. AWS CLI

### `Query`

다음 코드 예시에서는 `Query`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
#############################################################################
# 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 getopts 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
}
```

이 예제에 사용된 유틸리티 함수

```
###############################################################################
# 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
}
```

- 자세한 API 내용은 AWS CLI 명령 참조의 쿼리를 참조하세요.

### Scan

다음 코드 예시에서는 Scan을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
###############################################################################
# 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 getopts 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
}
```

이 예제에 사용된 유틸리티 함수

```
###############################################################################
# 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
}
```

- 자세한 API 내용은 명령 참조의 스캔을 참조하세요. AWS CLI

### `UpdateItem`

다음 코드 예시에서는 `UpdateItem`을 사용하는 방법을 보여 줍니다.

## AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
###############################################################################
# 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 getopts 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

}
```

이 예제에 사용된 유틸리티 함수

```
###############################################################################
# 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
}
```

- 자세한 API 내용은 명령 참조UpdateItem의 섹션을 참조하세요. AWS CLI

## 를 Bash 스크립트 AWS CLI 와 함께 사용하는 Amazon EC2 예제

다음 코드 예제에서는 Amazon 에서 Bash 스크립트 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다EC2.

기본 사항은 서비스 내에서 필수 작업을 수행하는 방법을 보여주는 코드 예제입니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 기본 사항
- 작업

### 기본 사항

기본 사항 알아보기

다음 코드 예시는 다음과 같은 작업을 수행하는 방법을 보여줍니다.

- 키 페어 및 보안 그룹을 생성합니다.
- Amazon Machine Image(AMI)와 호환되는 인스턴스 유형을 선택한 다음 인스턴스를 생성합니다.
- 인스턴스를 중지한 후 다시 시작합니다.
- 인스턴스와 탄력적 IP 주소 연결.
- 를 사용하여 인스턴스에 연결SSH한 다음 리소스를 정리합니다.

## AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

명령 프롬프트에서 대화형 시나리오를 실행합니다.

```
###############################################################################
# function get_started_with_ec2_instances
#
# Runs an interactive scenario that shows how to get started using EC2 instances.
#
#     "EC2 access" permissions are needed to run this code.
#
# Returns:
#       0 - If successful.
#       1 - If an error occurred.
###############################################################################
function get_started_with_ec2_instances() {
  # Requires version 4 for mapfile.
  local required_version=4.0

  # Get the current Bash version
  # Check if BASH_VERSION is set
  local current_version
  if [[ -n "$BASH_VERSION" ]]; then
    # Convert BASH_VERSION to a number for comparison
    current_version=$BASH_VERSION
  else
    # Get the current Bash version using the bash command
    current_version=$(bash --version | head -n 1 | awk '{ print $4 }')
  fi

  # Convert version strings to numbers for comparison
  local required_version_num current_version_num
  required_version_num=$(echo "$required_version" | awk -F. '{ print ($1 * 10000) +
($2 * 100) + $3 }')
  current_version_num=$(echo "$current_version" | awk -F. '{ print ($1 * 10000) +
($2 * 100) + $3 }')
```

```
 # Compare versions
 if ((current_version_num < required_version_num)); then
   echo "Error: This script requires Bash version $required_version or higher."
   echo "Your current Bash version is number is $current_version."
   exit 1
 fi

 {
   if [ "$EC2_OPERATIONS_SOURCED" != "True" ]; then

     source ./ec2_operations.sh
   fi
 }

 echo_repeat "*" 88
 echo "Welcome to the Amazon Elastic Compute Cloud (Amazon EC2) get started with
instances demo."
 echo_repeat "*" 88
 echo

 echo "Let's create an RSA key pair that you can be use to securely connect to "
 echo "your EC2 instance."

 echo -n "Enter a unique name for your key: "
 get_input
 local key_name
 key_name=$get_input_result

 local temp_dir
 temp_dir=$(mktemp -d)
 local key_file_name="$temp_dir/${key_name}.pem"

 if ec2_create_keypair -n "${key_name}" -f "${key_file_name}"; then
   echo "Created a key pair $key_name and saved the private key to $key_file_name"
   echo
 else
   errecho "The key pair failed to create. This demo will exit."
   return 1
 fi

 chmod 400 "${key_file_name}"

 if yes_no_input "Do you want to list some of your key pairs? (y/n) "; then
   local keys_and_fingerprints
```

```
    keys_and_fingerprints="$(ec2_describe_key_pairs)" && {
      local image_name_and_id
      while IFS=$'\n' read -r image_name_and_id; do
        local entries
        IFS=$'\t' read -ra entries <<<"$image_name_and_id"
        echo "Found rsa key ${entries[0]} with fingerprint:"
        echo "     ${entries[1]}"
      done <<<"$keys_and_fingerprints"

    }
  fi

  echo_repeat "*" 88
  echo_repeat "*" 88

  echo "Let's create a security group to manage access to your instance."
  echo -n "Enter a unique name for your security group: "
  get_input
  local security_group_name
  security_group_name=$get_input_result
  local security_group_id
  security_group_id=$(ec2_create_security_group -n "$security_group_name" \
    -d "Security group for EC2 instance") || {
    errecho "The security failed to create. This demo will exit."
    clean_up "$key_name" "$key_file_name"
    return 1
  }

  echo "Security group created with ID $security_group_id"
  echo

  local public_ip
  public_ip=$(curl -s http://checkip.amazonaws.com)

  echo "Let's add a rule to allow SSH only from your current IP address."
  echo "Your public IP address is $public_ip"
  echo -n "press return to add this rule to your security group."
  get_input

  if ! ec2_authorize_security_group_ingress -g "$security_group_id" -i "$public_ip"
-p tcp -f 22 -t 22; then
    errecho "The security group rules failed to update. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1
```

```
  fi

  echo "Security group rules updated"

  local security_group_description
  security_group_description="$(ec2_describe_security_groups -g
 "${security_group_id}")" || {
    errecho "Failed to describe security groups. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1
  }

  mapfile -t parameters <<<"$security_group_description"
  IFS=$'\t' read -ra entries <<<"${parameters[0]}"
  echo "Security group: ${entries[0]}"
  echo "    ID: ${entries[1]}"
  echo "    VPC: ${entries[2]}"
  echo "Inbound permissions:"
  IFS=$'\t' read -ra entries <<<"${parameters[1]}"
  echo "    IpProtocol: ${entries[0]}"
  echo "    FromPort: ${entries[1]}"
  echo "    ToPort: ${entries[2]}"
  echo "    CidrIp: ${parameters[2]}"

  local parameters
  parameters="$(ssm_get_parameters_by_path -p "/aws/service/ami-amazon-linux-
latest")" || {
    errecho "Failed to get parameters. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1

  }

  local image_ids=""
  mapfile -t parameters <<<"$parameters"
  for image_name_and_id in "${parameters[@]}"; do
    IFS=$'\t' read -ra values <<<"$image_name_and_id"
    if [[ "${values[0]}" == *"amzn2"* ]]; then
      image_ids+="${values[1]} "
    fi
  done

  local images
  images="$(ec2_describe_images -i "$image_ids")" || {
```

```
    errecho "Failed to describe images. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1

  }

  new_line_and_tab_to_list "$images"
  local images=("${list_result[@]}")

  # Get the size of the array
  local images_count=${#images[@]}

  if ((images_count == 0)); then
    errecho "No images found. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1
  fi

  echo_repeat "*" 88
  echo_repeat "*" 88

  echo "Let's create an instance from an Amazon Linux 2 AMI. Here are some options:"
  for ((i = 0; i < images_count; i += 3)); do
    echo "$(((i / 3) + 1)) - ${images[$i]}"
  done

  integer_input "Please enter the number of the AMI you want to use: " 1
"$((images_count / 3))"
  local choice=$get_input_result
  choice=$(((choice - 1) * 3))

  echo "Great choice."
  echo

  local architecture=${images[$((choice + 1))]}
  local image_id=${images[$((choice + 2))]}
  echo "Here are some instance types that support the ${architecture} architecture
of the image:"
  response="$(ec2_describe_instance_types -a "${architecture}" -t
"*.micro,*.small")" || {
    errecho "Failed to describe instance types. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1
  }
```

```
  local instance_types
  mapfile -t instance_types <<<"$response"

  # Get the size of the array
  local instance_types_count=${#instance_types[@]}

  echo "Here are some options:"
  for ((i = 0; i < instance_types_count; i++)); do
    echo "$((i + 1)) - ${instance_types[$i]}"
  done

  integer_input "Which one do you want to use? " 1 "${#instance_types[@]}
"
  choice=$get_input_result
  local instance_type=${instance_types[$((choice - 1))]}
  echo "Another great choice."
  echo

  echo "Creating your instance and waiting for it to start..."
  local instance_id
  instance_id=$(ec2_run_instances -i "$image_id" -t "$instance_type" -k "$key_name"
 -s "$security_group_id") || {
    errecho "Failed to run instance. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id"
    return 1
  }

  ec2_wait_for_instance_running -i "$instance_id"
  echo "Your instance is ready:"
  echo

  local instance_details
  instance_details="$(ec2_describe_instances -i "${instance_id}")"

  echo
  print_instance_details "${instance_details}"

  local public_ip
  public_ip=$(echo "${instance_details}" | awk '{print $6}')
  echo
  echo "You can use SSH to connect to your instance"
  echo "If the connection attempt times out, you might have to manually update the
 SSH ingress rule"
```

```
  echo "for your IP address in the AWS Management Console."
  connect_to_instance "$key_file_name" "$public_ip"

  echo -n "Press Enter when you're ready to continue the demo: "
  get_input

  echo_repeat "*" 88
  echo_repeat "*" 88

  echo "Let's stop and start your instance to see what changes."
  echo "Stopping your instance and waiting until it's stopped..."
  ec2_stop_instances -i "$instance_id"
  ec2_wait_for_instance_stopped -i "$instance_id"

  echo "Your instance is stopped. Restarting..."

  ec2_start_instances -i "$instance_id"
  ec2_wait_for_instance_running -i "$instance_id"

  echo "Your instance is running again."
  local instance_details
  instance_details="$(ec2_describe_instances -i "${instance_id}")"

  print_instance_details "${instance_details}"

  public_ip=$(echo "${instance_details}" | awk '{print $6}')

  echo "Every time your instance is restarted, its public IP address changes"
  connect_to_instance "$key_file_name" "$public_ip"

  echo -n "Press Enter when you're ready to continue the demo: "
  get_input

  echo_repeat "*" 88
  echo_repeat "*" 88

  echo "You can allocate an Elastic IP address and associate it with your instance"
  echo "to keep a consistent IP address even when your instance restarts."

  local result
  result=$(ec2_allocate_address -d vpc) || {
    errecho "Failed to allocate an address. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id" "$instance_id"
    return 1
```

```
  }

  local elastic_ip allocation_id
  elastic_ip=$(echo "$result" | awk '{print $1}')
  allocation_id=$(echo "$result" | awk '{print $2}')

  echo "Allocated static Elastic IP address: $elastic_ip"

  local association_id
  association_id=$(ec2_associate_address -i "$instance_id" -a "$allocation_id") || {
    errecho "Failed to associate an address. This demo will exit."
    clean_up "$key_name" "$key_file_name" "$security_group_id" "$instance_id"
"$allocation_id"
    return 1
  }

  echo "Associated your Elastic IP with your instance."
  echo "You can now use SSH to connect to your instance by using the Elastic IP."
  connect_to_instance "$key_file_name" "$elastic_ip"

  echo -n "Press Enter when you're ready to continue the demo: "
  get_input

  echo_repeat "*" 88
  echo_repeat "*" 88

  echo "Let's stop and start your instance to see what changes."
  echo "Stopping your instance and waiting until it's stopped..."
  ec2_stop_instances -i "$instance_id"
  ec2_wait_for_instance_stopped -i "$instance_id"

  echo "Your instance is stopped. Restarting..."

  ec2_start_instances -i "$instance_id"
  ec2_wait_for_instance_running -i "$instance_id"

  echo "Your instance is running again."
  local instance_details
  instance_details="$(ec2_describe_instances -i "${instance_id}")"

  print_instance_details "${instance_details}"

  echo "Because you have associated an Elastic IP with your instance, you can"
  echo "connect by using a consistent IP address after the instance restarts."
```

```
  connect_to_instance "$key_file_name" "$elastic_ip"

  echo -n "Press Enter when you're ready to continue the demo: "
  get_input

  echo_repeat "*" 88
  echo_repeat "*" 88

  if yes_no_input "Do you want to delete the resources created in this demo: (y/n)
 "; then
    clean_up "$key_name" "$key_file_name" "$security_group_id" "$instance_id" \
      "$allocation_id" "$association_id"
  else
    echo "The following resources were not deleted."
    echo "Key pair: $key_name"
    echo "Key file: $key_file_name"
    echo "Security group: $security_group_id"
    echo "Instance: $instance_id"
    echo "Elastic IP address: $elastic_ip"
  fi
}

###############################################################################
# function clean_up
#
# This function cleans up the created resources.
#     $1 - The name of the ec2 key pair to delete.
#     $2 - The name of the key file to delete.
#     $3 - The ID of the security group to delete.
#     $4 - The ID of the instance to terminate.
#     $5 - The ID of the elastic IP address to release.
#     $6 - The ID of the elastic IP address to disassociate.
#
# Returns:
#       0 - If successful.
#       1 - If an error occurred.
###############################################################################
function clean_up() {
  local result=0
  local key_pair_name=$1
  local key_file_name=$2
  local security_group_id=$3
  local instance_id=$4
  local allocation_id=$5
```

```
  local association_id=$6

  if [ -n "$association_id" ]; then
    # bashsupport disable=BP2002
    if (ec2_disassociate_address -a "$association_id"); then
      echo "Disassociated elastic IP address with ID $association_id"
    else
      errecho "The elastic IP address disassociation failed."
      result=1
    fi
  fi

  if [ -n "$allocation_id" ]; then
    # bashsupport disable=BP2002
    if (ec2_release_address -a "$allocation_id"); then
      echo "Released elastic IP address with ID $allocation_id"
    else
      errecho "The elastic IP address release failed."
      result=1
    fi
  fi

  if [ -n "$instance_id" ]; then
    # bashsupport disable=BP2002
    if (ec2_terminate_instances -i "$instance_id"); then
      echo "Started terminating instance with ID $instance_id"

      ec2_wait_for_instance_terminated -i "$instance_id"
    else
      errecho "The instance terminate failed."
      result=1
    fi
  fi

  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 getopts 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
}
```

이 시나리오에 사용된 DynamoDB 함수입니다.

```
###############################################################################
# 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 getopts 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 getopts "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 getopts 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 getopts 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 getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_authorize_security_group_ingress"
    echo "Authorizes an ingress rule for an Amazon Elastic Compute Cloud (Amazon
 EC2) security group."
    echo "  -g security_group_id - The ID of the security group."
    echo "  -i ip_address - The IP address or CIDR block to authorize."
    echo "  -p protocol - The protocol to authorize (e.g., tcp, udp, icmp)."
    echo "  -f from_port - The start of the port range to authorize."
    echo "  -t to_port - The end of the port range to authorize."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "g:i:p:f:t:h" option; do
    case "${option}" in
      g) security_group_id="${OPTARG}" ;;
      i) ip_address="${OPTARG}" ;;
      p) protocol="${OPTARG}" ;;
      f) from_port="${OPTARG}" ;;
      t) to_port="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
```

```
    esac
  done
  export OPTIND=1

  if [[ -z "$security_group_id" ]]; then
    errecho "ERROR: You must provide a security group ID with the -g parameter."
    usage
    return 1
  fi

  if [[ -z "$ip_address" ]]; then
    errecho "ERROR: You must provide an IP address or CIDR block with the -i
 parameter."
    usage
    return 1
  fi

  if [[ -z "$protocol" ]]; then
    errecho "ERROR: You must provide a protocol with the -p parameter."
    usage
    return 1
  fi

  if [[ -z "$from_port" ]]; then
    errecho "ERROR: You must provide a start port with the -f parameter."
    usage
    return 1
  fi

  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 getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_describe_images"
    echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2) images."
    echo "  -i image_ids - A space-separated list of image IDs (optional)."
    echo "  -h - Display help."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "i:h" option; do
    case "${option}" in
      i) image_ids="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
```

```
        return 1
        ;;
    esac
  done
  export OPTIND=1

  local aws_cli_args=()

  if [[ -n "$image_ids" ]]; then
    # shellcheck disable=SC2206
    aws_cli_args+=("--image-ids" $image_ids)
  fi

  response=$(aws ec2 describe-images \
    "${aws_cli_args[@]}" \
    --query 'Images[*].[Description,Architecture,ImageId]' \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports describe-images operation failed.$response"
    return 1
  }

  echo "$response"

  return 0
}

################################################################################
# ec2_describe_instance_types
#
# This function describes EC2 instance types filtered by processor architecture
# and optionally by instance type. It takes the following arguments:
#
# -a, --architecture ARCHITECTURE  Specify the processor architecture (e.g., x86_64)
# -t, --type INSTANCE_TYPE         Comma-separated list of instance types (e.g.,
 t2.micro)
# -h, --help                       Show the usage help
#
# The function prints the instance type and supported architecture for each
# matching instance type.
################################################################################
function ec2_describe_instance_types() {
  local architecture=""
  local instance_types=""
```

```
  # bashsupport disable=BP5008
  function usage() {
    echo "Usage: ec2_describe_instance_types [-a|--architecture ARCHITECTURE] [-t|--
type INSTANCE_TYPE] [-h|--help]"
    echo "  -a, --architecture ARCHITECTURE  Specify the processor architecture
 (e.g., x86_64)"
    echo "  -t, --type INSTANCE_TYPE         Comma-separated list of instance types
 (e.g., t2.micro)"
    echo "  -h, --help                       Show this help message"
  }

  while [[ $# -gt 0 ]]; do
    case "$1" in
      -a | --architecture)
        architecture="$2"
        shift 2
        ;;
      -t | --type)
        instance_types="$2"
        shift 2
        ;;
      -h | --help)
        usage
        return 0
        ;;
      *)
        echo "Unknown argument: $1"
        return 1
        ;;
    esac
  done

  if [[ -z "$architecture" ]]; then
    errecho "Error: Architecture not specified."
    usage
    return 1
  fi

  if [[ -z "$instance_types" ]]; then
    errecho "Error: Instance type not specified."
    usage
    return 1
  fi
```

```
local tmp_json_file="temp_ec2.json"
echo -n '[
  {
    "Name": "processor-info.supported-architecture",
    "Values": [' >"$tmp_json_file"

local items
IFS=',' read -ra items <<<"$architecture"
local array_size
array_size=${#items[@]}
for i in $(seq 0 $((array_size - 1))); do
  echo -n '"'"${items[$i]}"'"' >>"$tmp_json_file"
  if [[ $i -lt $((array_size - 1)) ]]; then
    echo -n ',' >>"$tmp_json_file"
  fi
done
echo -n ']},
  {
  "Name": "instance-type",
    "Values": [' >>"$tmp_json_file"
IFS=',' read -ra items <<<"$instance_types"
local array_size
array_size=${#items[@]}
for i in $(seq 0 $((array_size - 1))); do
  echo -n '"'"${items[$i]}"'"' >>"$tmp_json_file"
  if [[ $i -lt $((array_size - 1)) ]]; then
    echo -n ',' >>"$tmp_json_file"
  fi
done

echo -n ']}]' >>"$tmp_json_file"

local response
response=$(aws ec2 describe-instance-types --filters file://"$tmp_json_file" \
  --query 'InstanceTypes[*].[InstanceType]' --output text)

local error_code=$?

rm "$tmp_json_file"

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  echo "ERROR: AWS reports describe-instance-types operation failed."
```

```
    return 1
  fi

  echo "$response"
  return 0
}

###############################################################################
# function ec2_run_instances
#
# This function launches one or more Amazon Elastic Compute Cloud (Amazon EC2)
 instances.
#
# Parameters:
#       -i image_id - The ID of the Amazon Machine Image (AMI) to use.
#       -t instance_type - The instance type to use (e.g., t2.micro).
#       -k key_pair_name - The name of the key pair to use.
#       -s security_group_id - The ID of the security group to use.
#       -c count - The number of instances to launch (default: 1).
#       -h - Display help.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_run_instances() {
  local image_id instance_type key_pair_name security_group_id count response
  local option OPTARG # Required to use getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_run_instances"
    echo "Launches one or more Amazon Elastic Compute Cloud (Amazon EC2) instances."
    echo "  -i image_id - The ID of the Amazon Machine Image (AMI) to use."
    echo "  -t instance_type - The instance type to use (e.g., t2.micro)."
    echo "  -k key_pair_name - The name of the key pair to use."
    echo "  -s security_group_id - The ID of the security group to use."
    echo "  -c count - The number of instances to launch (default: 1)."
    echo "  -h - Display help."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "i:t:k:s:c:h" option; do
```

```
    case "${option}" in
      i) image_id="${OPTARG}" ;;
      t) instance_type="${OPTARG}" ;;
      k) key_pair_name="${OPTARG}" ;;
      s) security_group_id="${OPTARG}" ;;
      c) count="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$image_id" ]]; then
    errecho "ERROR: You must provide an Amazon Machine Image (AMI) ID with the -i
parameter."
    usage
    return 1
  fi

  if [[ -z "$instance_type" ]]; then
    errecho "ERROR: You must provide an instance type with the -t parameter."
    usage
    return 1
  fi

  if [[ -z "$key_pair_name" ]]; then
    errecho "ERROR: You must provide a key pair name with the -k parameter."
    usage
    return 1
  fi

  if [[ -z "$security_group_id" ]]; then
    errecho "ERROR: You must provide a security group ID with the -s parameter."
    usage
    return 1
  fi
```

```
  if [[ -z "$count" ]]; then
    count=1
  fi

  response=$(aws ec2 run-instances \
    --image-id "$image_id" \
    --instance-type "$instance_type" \
    --key-name "$key_pair_name" \
    --security-group-ids "$security_group_id" \
    --count "$count" \
    --query 'Instances[*].[InstanceId]' \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports run-instances operation failed.$response"
    return 1
  }

  echo "$response"

  return 0
}

################################################################################
# function ec2_describe_instances
#
# This function describes one or more Amazon Elastic Compute Cloud (Amazon EC2)
 instances.
#
# Parameters:
#       -i instance_id - The ID of the instance to describe (optional).
#       -q query - The query to filter the response (optional).
#       -h - Display help.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
################################################################################
function ec2_describe_instances() {
  local instance_id query response
  local option OPTARG # Required to use getopts 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 getopts "i:q:h" option; do
    case "${option}" in
      i) instance_id="${OPTARG}" ;;
      q) query="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  local aws_cli_args=()

  if [[ -n "$instance_id" ]]; then
    # shellcheck disable=SC2206
    aws_cli_args+=("--instance-ids" $instance_id)
  fi

  local query_arg=""
  if [[ -n "$query" ]]; then
    query_arg="--query '$query'"
  else
    query_arg="--query Reservations[*].Instances[*].
[InstanceId,ImageId,InstanceType,KeyName,VpcId,PublicIpAddress,State.Name]"
  fi

  # shellcheck disable=SC2086
  response=$(aws ec2 describe-instances \
    "${aws_cli_args[@]}" \
```

```
    $query_arg \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports describe-instances operation failed.$response"
    return 1
  }

  echo "$response"

  return 0
}

###############################################################################
# function ec2_stop_instances
#
# This function stops one or more Amazon Elastic Compute Cloud (Amazon EC2)
 instances.
#
# Parameters:
#       -i instance_id - The ID(s) of the instance(s) to stop (comma-separated).
#       -h - Display help.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_stop_instances() {
  local instance_ids
  local option OPTARG # Required to use getopts 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 getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_start_instances"
    echo "Starts one or more Amazon Elastic Compute Cloud (Amazon EC2) instances."
    echo "  -i instance_id - The ID(s) of the instance(s) to start (comma-
separated)."
    echo "  -h - Display help."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "i:h" option; do
    case "${option}" in
      i) instance_ids="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$instance_ids" ]]; then
    errecho "ERROR: You must provide one or more instance IDs with the -i
 parameter."
    usage
    return 1
  fi

  response=$(aws ec2 start-instances \
    --instance-ids "${instance_ids}") || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports start-instances operation failed with $response."
    return 1
  }
```

```
  return 0
}

################################################################################
# function ec2_allocate_address
#
# This function allocates an Elastic IP address for use with Amazon Elastic Compute
 Cloud (Amazon EC2) instances in a specific AWS Region.
#
# Parameters:
#       -d domain - The domain for the Elastic IP address (either 'vpc' or
 'standard').
#
# Returns:
#       The allocated Elastic IP address, or an error message if the operation
 fails.
# And:
#       0 - If successful.
#       1 - If it fails.
#
################################################################################
function ec2_allocate_address() {
  local domain response

  # Function to display usage information
  function usage() {
    echo "function ec2_allocate_address"
    echo "Allocates an Elastic IP address for use with Amazon Elastic Compute Cloud
 (Amazon EC2) instances in a specific AWS Region."
    echo "  -d domain - The domain for the Elastic IP address (either 'vpc' or
 'standard')."
    echo ""
  }

  # Parse the command-line arguments
  while getopts "d:h" option; do
    case "${option}" in
      d) domain="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
```

```
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  # Validate the input parameters
  if [[ -z "$domain" ]]; then
    errecho "ERROR: You must provide a domain with the -d parameter (either 'vpc' or
 'standard')."
    return 1
  fi

  if [[ "$domain" != "vpc" && "$domain" != "standard" ]]; then
    errecho "ERROR: Invalid domain value. Must be either 'vpc' or 'standard'."
    return 1
  fi

  # Allocate the Elastic IP address
  response=$(aws ec2 allocate-address \
    --domain "$domain" \
    --query "[PublicIp,AllocationId]" \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports allocate-address operation failed."
    errecho "$response"
    return 1
  }

  echo "$response"
  return 0
}

###############################################################################
# function ec2_associate_address
#
# This function associates an Elastic IP address with an Amazon Elastic Compute
 Cloud (Amazon EC2) instance.
#
# Parameters:
#       -a allocation_id - The allocation ID of the Elastic IP address to associate.
```

```
#       -i instance_id - The ID of the EC2 instance to associate the Elastic IP
 address with.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
#
###############################################################################
function ec2_associate_address() {
  local allocation_id instance_id response

  # Function to display usage information
  function usage() {
    echo "function ec2_associate_address"
    echo "Associates an Elastic IP address with an Amazon Elastic Compute Cloud
 (Amazon EC2) instance."
    echo "  -a allocation_id - The allocation ID of the Elastic IP address to
 associate."
    echo "  -i instance_id - The ID of the EC2 instance to associate the Elastic IP
 address with."
    echo ""
  }

  # Parse the command-line arguments
  while getopts "a:i:h" option; do
    case "${option}" in
      a) allocation_id="${OPTARG}" ;;
      i) instance_id="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  # Validate the input parameters
  if [[ -z "$allocation_id" ]]; then
    errecho "ERROR: You must provide an allocation ID with the -a parameter."
```

```
    return 1
  fi

  if [[ -z "$instance_id" ]]; then
    errecho "ERROR: You must provide an instance ID with the -i parameter."
    return 1
  fi

  # Associate the Elastic IP address
  response=$(aws ec2 associate-address \
    --allocation-id "$allocation_id" \
    --instance-id "$instance_id" \
    --query "AssociationId" \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports associate-address operation failed."
    errecho "$response"
    return 1
  }

  echo "$response"
  return 0
}

################################################################################
# function ec2_disassociate_address
#
# This function disassociates an Elastic IP address from an Amazon Elastic Compute
 Cloud (Amazon EC2) instance.
#
# Parameters:
#       -a association_id - The association ID that represents the association of
 the Elastic IP address with an instance.
#
# And:
#       0 - If successful.
#       1 - If it fails.
#
################################################################################
function ec2_disassociate_address() {
  local association_id response

  # Function to display usage information
  function usage() {
```

```
    echo "function ec2_disassociate_address"
    echo "Disassociates an Elastic IP address from an Amazon Elastic Compute Cloud
 (Amazon EC2) instance."
    echo "  -a association_id - The association ID that represents the association
 of the Elastic IP address with an instance."
    echo ""
  }

  # Parse the command-line arguments
  while getopts "a:h" option; do
    case "${option}" in
      a) association_id="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  # Validate the input parameters
  if [[ -z "$association_id" ]]; then
    errecho "ERROR: You must provide an association ID with the -a parameter."
    return 1
  fi

  response=$(aws ec2 disassociate-address \
    --association-id "$association_id") || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports disassociate-address operation failed."
    errecho "$response"
    return 1
  }

  return 0
}

################################################################################
# function ec2_release_address
```

```
#
# This function releases an Elastic IP address from an Amazon Elastic Compute Cloud
 (Amazon EC2) instance.
#
# Parameters:
#       -a allocation_id - The allocation ID of the Elastic IP address to release.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
#
###############################################################################
function ec2_release_address() {
  local allocation_id response

  # Function to display usage information
  function usage() {
    echo "function ec2_release_address"
    echo "Releases an Elastic IP address from an Amazon Elastic Compute Cloud
 (Amazon EC2) instance."
    echo "  -a allocation_id - The allocation ID of the Elastic IP address to
 release."
    echo ""
  }

  # Parse the command-line arguments
  while getopts "a:h" option; do
    case "${option}" in
      a) allocation_id="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  # Validate the input parameters
  if [[ -z "$allocation_id" ]]; then
```

```
    errecho "ERROR: You must provide an allocation ID with the -a parameter."
    return 1
  fi

  response=$(aws ec2 release-address \
    --allocation-id "$allocation_id") || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports release-address operation failed."
    errecho "$response"
    return 1
  }

  return 0
}

###############################################################################
# function ec2_terminate_instances
#
# This function terminates one or more Amazon Elastic Compute Cloud (Amazon EC2)
# instances using the AWS CLI.
#
# Parameters:
#       -i instance_ids - A space-separated list of instance IDs.
#       -h - Display help.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_terminate_instances() {
  local instance_ids response
  local option OPTARG # Required to use getopts 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 getopts 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 getopts "i:h" option; do
    case "${option}" in
      i) security_group_id="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$security_group_id" ]]; then
    errecho "ERROR: You must provide a security group ID with the -i parameter."
    usage
    return 1
  fi

  response=$(aws ec2 delete-security-group --group-id "$security_group_id" --output
 text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports delete-security-group operation failed.$response"
    return 1
```

```
  }

  return 0
}

###############################################################################
# function ec2_delete_keypair
#
# This function deletes an Amazon EC2 ED25519 or 2048-bit RSA key pair.
#
# Parameters:
#       -n key_pair_name - A key pair name.
#
# And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function ec2_delete_keypair() {
  local key_pair_name response

  local option OPTARG # Required to use getopts 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 getopts "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
}
```

이 시나리오에 사용된 유틸리티 함수입니다.

```
###############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
###############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

###############################################################################
# function aws_cli_error_log()
#
# This function is used to log the error messages from the AWS CLI.
#
# The function expects the following argument:
#         $1 - The error code returned by the AWS CLI.
#
#  Returns:
#          0: - Success.
#
###############################################################################
function aws_cli_error_log() {
```

```
  local err_code=$1
  errecho "Error code : $err_code"
  if [ "$err_code" == 1 ]; then
    errecho "  One or more S3 transfers failed."
  elif [ "$err_code" == 2 ]; then
    errecho "  Command line failed to parse."
  elif [ "$err_code" == 130 ]; then
    errecho "  Process received SIGINT."
  elif [ "$err_code" == 252 ]; then
    errecho "  Command syntax invalid."
  elif [ "$err_code" == 253 ]; then
    errecho "  The system environment or configuration was invalid."
  elif [ "$err_code" == 254 ]; then
    errecho "  The service returned an error."
  elif [ "$err_code" == 255 ]; then
    errecho "  255 is a catch-all error."
  fi

  return 0
}
```

- API 자세한 내용은 AWS CLI 명령 참조의 다음 주제를 참조하세요.
  - AllocateAddress
  - AssociateAddress
  - AuthorizeSecurityGroupIngress
  - CreateKeyPair
  - CreateSecurityGroup
  - DeleteKeyPair
  - DeleteSecurityGroup
  - DescribeImages
  - DescribeInstanceTypes
  - DescribeInstances
  - DescribeKeyPairs
  - DescribeSecurityGroups
  - DisassociateAddress
  - ReleaseAddress

- RunInstances
- StartInstances
- StopInstances
- TerminateInstances
- UnmonitorInstances

## 작업

### `AllocateAddress`

다음 코드 예시에서는 `AllocateAddress`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
###############################################################################
# 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
}
```

이 예제에 사용된 유틸리티 함수

```
##############################################################################
# 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
}
```

- 자세한 API 내용은 명령 참조AllocateAddress의 섹션을 참조하세요. AWS CLI

### `AssociateAddress`

다음 코드 예시에서는 `AssociateAddress`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
#############################################################################
# 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
}
```

이 예제에 사용된 유틸리티 함수

```
#############################################################################
# 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
}
```

- 자세한 API 내용은 명령 참조AssociateAddress의 섹션을 참조하세요. AWS CLI

### `AuthorizeSecurityGroupIngress`

다음 코드 예시에서는 `AuthorizeSecurityGroupIngress`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
#############################################################################
# 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 getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_authorize_security_group_ingress"
    echo "Authorizes an ingress rule for an Amazon Elastic Compute Cloud (Amazon
 EC2) security group."
    echo "  -g security_group_id - The ID of the security group."
    echo "  -i ip_address - The IP address or CIDR block to authorize."
    echo "  -p protocol - The protocol to authorize (e.g., tcp, udp, icmp)."
    echo "  -f from_port - The start of the port range to authorize."
    echo "  -t to_port - The end of the port range to authorize."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "g:i:p:f:t:h" option; do
    case "${option}" in
      g) security_group_id="${OPTARG}" ;;
      i) ip_address="${OPTARG}" ;;
      p) protocol="${OPTARG}" ;;
      f) from_port="${OPTARG}" ;;
      t) to_port="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
```

```
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$security_group_id" ]]; then
    errecho "ERROR: You must provide a security group ID with the -g parameter."
    usage
    return 1
  fi

  if [[ -z "$ip_address" ]]; then
    errecho "ERROR: You must provide an IP address or CIDR block with the -i
parameter."
    usage
    return 1
  fi

  if [[ -z "$protocol" ]]; then
    errecho "ERROR: You must provide a protocol with the -p parameter."
    usage
    return 1
  fi

  if [[ -z "$from_port" ]]; then
    errecho "ERROR: You must provide a start port with the -f parameter."
    usage
    return 1
  fi

  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
}
```

이 예제에 사용된 유틸리티 함수

```
##############################################################################
# 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
}
```

- 자세한 API 내용은 명령 참조AuthorizeSecurityGroupIngress의 섹션을 참조하세요. AWS CLI

### `CreateKeyPair`

다음 코드 예시에서는 `CreateKeyPair`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
###############################################################################
# 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 getopts 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 getopts "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
}
```

이 예제에 사용된 유틸리티 함수

```
##############################################################################
# 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
}
```

- 자세한 API 내용은 명령 참조CreateKeyPair의 섹션을 참조하세요. AWS CLI

### `CreateSecurityGroup`

다음 코드 예시에서는 `CreateSecurityGroup`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
#############################################################################
# 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
}
```

이 예제에 사용된 유틸리티 함수

```
################################################################################
# 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
}
```

- 자세한 API 내용은 명령 참조CreateSecurityGroup의 섹션을 참조하세요. AWS CLI

### `DeleteKeyPair`

다음 코드 예시에서는 `DeleteKeyPair`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
###############################################################################
```

```
# 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 getopts 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 getopts "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
}
```

이 예제에 사용된 유틸리티 함수

```
##############################################################################
# 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
}
```

- 자세한 API 내용은 명령 참조DeleteKeyPair의 섹션을 참조하세요. AWS CLI

### `DeleteSecurityGroup`

다음 코드 예시에서는 `DeleteSecurityGroup`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
###############################################################################
# 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 getopts 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 getopts "i:h" option; do
    case "${option}" in
      i) security_group_id="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$security_group_id" ]]; then
    errecho "ERROR: You must provide a security group ID with the -i parameter."
    usage
    return 1
  fi

  response=$(aws ec2 delete-security-group --group-id "$security_group_id" --output
 text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports delete-security-group operation failed.$response"
    return 1
  }

  return 0
}
```

이 예제에 사용된 유틸리티 함수

```
###############################################################################
# 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
}
```

- 자세한 API 내용은 명령 참조DeleteSecurityGroup의 섹션을 참조하세요. AWS CLI

### `DescribeImages`

다음 코드 예시에서는 DescribeImages을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
##############################################################################
# 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 getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_describe_images"
    echo "Describes one or more Amazon Elastic Compute Cloud (Amazon EC2) images."
    echo "  -i image_ids - A space-separated list of image IDs (optional)."
```

```
    echo "  -h - Display help."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "i:h" option; do
    case "${option}" in
      i) image_ids="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  local aws_cli_args=()

  if [[ -n "$image_ids" ]]; then
    # shellcheck disable=SC2206
    aws_cli_args+=("--image-ids" $image_ids)
  fi

  response=$(aws ec2 describe-images \
    "${aws_cli_args[@]}" \
    --query 'Images[*].[Description,Architecture,ImageId]' \
    --output text) || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports describe-images operation failed.$response"
    return 1
  }

  echo "$response"

  return 0
}
```

이 예제에 사용된 유틸리티 함수

```
#############################################################################
# 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
}
```

- 자세한 API 내용은 명령 참조DescribeImages의 섹션을 참조하세요. AWS CLI

### `DescribeInstanceTypes`

다음 코드 예시에서는 `DescribeInstanceTypes`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
###############################################################################
# ec2_describe_instance_types
#
# This function describes EC2 instance types filtered by processor architecture
# and optionally by instance type. It takes the following arguments:
#
# -a, --architecture ARCHITECTURE  Specify the processor architecture (e.g., x86_64)
# -t, --type INSTANCE_TYPE         Comma-separated list of instance types (e.g.,
 t2.micro)
# -h, --help                       Show the usage help
#
# The function prints the instance type and supported architecture for each
# matching instance type.
###############################################################################
function ec2_describe_instance_types() {
  local architecture=""
  local instance_types=""

  # bashsupport disable=BP5008
  function usage() {
    echo "Usage: ec2_describe_instance_types [-a|--architecture ARCHITECTURE] [-t|--
type INSTANCE_TYPE] [-h|--help]"
    echo "  -a, --architecture ARCHITECTURE  Specify the processor architecture
 (e.g., x86_64)"
    echo "  -t, --type INSTANCE_TYPE         Comma-separated list of instance types
 (e.g., t2.micro)"
```

```
  echo "  -h, --help                        Show this help message"
}

while [[ $# -gt 0 ]]; do
  case "$1" in
    -a | --architecture)
      architecture="$2"
      shift 2
      ;;
    -t | --type)
      instance_types="$2"
      shift 2
      ;;
    -h | --help)
      usage
      return 0
      ;;
    *)
      echo "Unknown argument: $1"
      return 1
      ;;
  esac
done

if [[ -z "$architecture" ]]; then
  errecho "Error: Architecture not specified."
  usage
  return 1
fi

if [[ -z "$instance_types" ]]; then
  errecho "Error: Instance type not specified."
  usage
  return 1
fi

local tmp_json_file="temp_ec2.json"
echo -n '[
  {
    "Name": "processor-info.supported-architecture",
    "Values": [' >"$tmp_json_file"

local items
IFS=',' read -ra items <<<"$architecture"
```

```
  local array_size
  array_size=${#items[@]}
  for i in $(seq 0 $((array_size - 1))); do
    echo -n '"'"${items[$i]}"'"' >>"$tmp_json_file"
    if [[ $i -lt $((array_size - 1)) ]]; then
      echo -n ',' >>"$tmp_json_file"
    fi
  done
  echo -n ']},
    {
    "Name": "instance-type",
      "Values": [' >>"$tmp_json_file"
  IFS=',' read -ra items <<<"$instance_types"
  local array_size
  array_size=${#items[@]}
  for i in $(seq 0 $((array_size - 1))); do
    echo -n '"'"${items[$i]}"'"' >>"$tmp_json_file"
    if [[ $i -lt $((array_size - 1)) ]]; then
      echo -n ',' >>"$tmp_json_file"
    fi
  done

  echo -n ']}]' >>"$tmp_json_file"

  local response
  response=$(aws ec2 describe-instance-types --filters file://"$tmp_json_file" \
    --query 'InstanceTypes[*].[InstanceType]' --output text)

  local error_code=$?

  rm "$tmp_json_file"

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    echo "ERROR: AWS reports describe-instance-types operation failed."
    return 1
  fi

  echo "$response"
  return 0
}
```

이 예제에 사용된 유틸리티 함수

```
#############################################################################
# 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
}
```

- 자세한 API 내용은 명령 참조DescribeInstanceTypes의 섹션을 참조하세요. AWS CLI

### `DescribeInstances`

다음 코드 예시에서는 `DescribeInstances`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
##############################################################################
# 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 getopts 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 getopts "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
}
```

이 예제에 사용된 유틸리티 함수

```
################################################################################
# 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
}
```

- 자세한 API 내용은 명령 참조DescribeInstances의 섹션을 참조하세요. AWS CLI

### `DescribeKeyPairs`

다음 코드 예시에서는 `DescribeKeyPairs`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
#############################################################################
# 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 getopts 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
}
```

이 예제에 사용된 유틸리티 함수

```
#############################################################################
# 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
}
```

- 자세한 API 내용은 명령 참조DescribeKeyPairs의 섹션을 참조하세요. AWS CLI

### `DescribeSecurityGroups`

다음 코드 예시에서는 `DescribeSecurityGroups`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
#############################################################################
# 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 getopts 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
}
```

이 예제에 사용된 유틸리티 함수

```
##############################################################################
```

```
# 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
}
```

- 자세한 API 내용은 명령 참조DescribeSecurityGroups의 섹션을 참조하세요. AWS CLI

### `DisassociateAddress`

다음 코드 예시에서는 `DisassociateAddress`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
###############################################################################
# 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
}
```

이 예제에 사용된 유틸리티 함수

```
########################################################################
# 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
}
```

- 자세한 API 내용은 명령 참조DisassociateAddress의 섹션을 참조하세요. AWS CLI

### `ReleaseAddress`

다음 코드 예시에서는 `ReleaseAddress`을 사용하는 방법을 보여 줍니다.

## AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
##############################################################################
# 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
}
```

이 예제에 사용된 유틸리티 함수

```
##############################################################################
# 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
}
```

- 자세한 API 내용은 명령 참조ReleaseAddress의 섹션을 참조하세요. AWS CLI

### `RunInstances`

다음 코드 예시에서는 `RunInstances`을 사용하는 방법을 보여 줍니다.

## AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
#############################################################################
# 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 getopts 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
}
```

이 예제에 사용된 유틸리티 함수

```
################################################################################
# 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
}
```

- 자세한 API 내용은 명령 참조RunInstances의 섹션을 참조하세요. AWS CLI

### `StartInstances`

다음 코드 예시에서는 `StartInstances`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
###############################################################################
# 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 getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function ec2_start_instances"
    echo "Starts one or more Amazon Elastic Compute Cloud (Amazon EC2) instances."
    echo "  -i instance_id - The ID(s) of the instance(s) to start (comma-
separated)."
    echo "  -h - Display help."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "i:h" option; do
    case "${option}" in
      i) instance_ids="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
```

```
  export OPTIND=1

  if [[ -z "$instance_ids" ]]; then
    errecho "ERROR: You must provide one or more instance IDs with the -i
 parameter."
    usage
    return 1
  fi

  response=$(aws ec2 start-instances \
    --instance-ids "${instance_ids}") || {
    aws_cli_error_log ${?}
    errecho "ERROR: AWS reports start-instances operation failed with $response."
    return 1
  }

  return 0
}
```

이 예제에 사용된 유틸리티 함수

```
##############################################################################
# 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
}
```

- 자세한 API 내용은 명령 참조StartInstances의 섹션을 참조하세요. AWS CLI

### `StopInstances`

다음 코드 예시에서는 StopInstances을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
#############################################################################
# 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 getopts 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
}
```

이 예제에 사용된 유틸리티 함수

```
#############################################################################
# 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
}
```

- 자세한 API 내용은 명령 참조StopInstances의 섹션을 참조하세요. AWS CLI

### `TerminateInstances`

다음 코드 예시에서는 `TerminateInstances`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
##############################################################################
# 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 getopts 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
}
```

이 예제에 사용된 유틸리티 함수

```
################################################################################
# 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
}
```

- 자세한 API 내용은 명령 참조TerminateInstances의 섹션을 참조하세요. AWS CLI

## HealthImaging 를 Bash 스크립트 AWS CLI 와 함께 사용하는 예제

다음 코드 예제에서는 와 AWS Command Line Interface 함께 Bash 스크립트를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 HealthImaging.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

### 작업

#### `CreateDatastore`

다음 코드 예시에서는 `CreateDatastore`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

```
###############################################################################
# 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 getopts 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 getopts "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
    errecho "ERROR: AWS reports medical-imaging create-datastore operation failed.
$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

- 자세한 API 내용은 명령 참조CreateDatastore의 섹션을 참조하세요. AWS CLI

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

### DeleteDatastore

다음 코드 예시에서는 DeleteDatastore을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

```
##############################################################################
# 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 getopts 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 getopts "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
}
```

- 자세한 API 내용은 명령 참조DeleteDatastore의 섹션을 참조하세요. AWS CLI

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

### GetDatastore

다음 코드 예시에서는 GetDatastore을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

```
##############################################################################
# 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 getopts 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 getopts "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
}
```

- 자세한 API 내용은 명령 참조GetDatastore의 섹션을 참조하세요. AWS CLI

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

### `ListDatastores`

다음 코드 예시에서는 `ListDatastores`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

```
###############################################################################
# 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 getopts 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 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=$(aws medical-imaging list-datastores \
    --output text \
    --query "datastoreSummaries[*][datastoreName, datastoreId, datastoreStatus]")
  error_code=${?}

  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
}
```

- 자세한 API 내용은 명령 참조ListDatastores의 섹션을 참조하세요. AWS CLI

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

# IAM 를 Bash 스크립트 AWS CLI 와 함께 사용하는 예제

다음 코드 예제에서는 와 AWS Command Line Interface 함께 Bash 스크립트를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다IAM.

기본 사항은 서비스 내에서 필수 작업을 수행하는 방법을 보여주는 코드 예제입니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 기본 사항

기본 사항 알아보기

다음 코드 예제에서는 사용자를 생성하고 역할을 수임하는 방법을 보여줍니다.

> **Warning**
>
> 보안 위험을 방지하려면 특별히 제작된 소프트웨어를 개발하거나 실제 데이터로 작업할 때 IAM 사용자를 인증에 사용하지 마세요. 대신 AWS IAM Identity Center과 같은 보안 인증 공급자를 통한 페더레이션을 사용하십시오.

- 권한이 없는 사용자를 생성합니다.
- 계정에 대한 Amazon S3 버킷을 나열할 수 있는 권한을 부여하는 역할을 생성합니다.

- 사용자가 역할을 수임할 수 있도록 정책을 추가합니다.
- 역할을 수임하고 임시 보안 인증 정보를 사용하여 S3 버킷을 나열한 후 리소스를 정리합니다.

### AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
###############################################################################
# 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
}
```

이 시나리오에서 사용되는 IAM 함수입니다.

```
###############################################################################
# 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 getopts 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 getopts "u:h" option; do
    case "${option}" in
      u) user_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    User name:   $user_name"
  iecho ""
```

```
  # If the user 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 getopts command in a function.

  # bashsupport disable=BP5008
```

```
function usage() {
  echo "function iam_create_user_access_key"
  echo "Creates an AWS Identity and Access Management (IAM) key pair."
  echo "  -u user_name   The name of the IAM user."
  echo "  [-f file_name]   Optional file name for the access key output."
  echo ""
}

# Retrieve the calling parameters.
while getopts "u:f:h" option; do
  case "${option}" in
    u) user_name="${OPTARG}" ;;
    f) file_name="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$user_name" ]]; then
  errecho "ERROR: You must provide a username with the -u parameter."
  usage
  return 1
fi

response=$(aws iam create-access-key \
  --user-name "$user_name" \
  --output text)

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
  errecho "ERROR: AWS reports create-access-key operation failed.$response"
  return 1
fi
```

```
  if [[ -n "$file_name" ]]; then
    echo "$response" >"$file_name"
  fi

  local key_id key_secret
  # shellcheck disable=SC2086
  key_id=$(echo $response | cut -f 2 -d ' ')
  # shellcheck disable=SC2086
  key_secret=$(echo $response | cut -f 4 -d ' ')

  echo "$key_id $key_secret"

  return 0
}

###############################################################################
# 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 getopts 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 getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_policy"
    echo "Creates an AWS Identity and Access Management (IAM) policy."
    echo "  -n policy_name   The name of the IAM policy."
    echo "  -p policy_json -- The policy document."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:p:h" option; do
    case "${option}" in
      n) policy_name="${OPTARG}" ;;
      p) policy_document="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
```

```
    esac
  done
  export OPTIND=1

  if [[ -z "$policy_name" ]]; then
    errecho "ERROR: You must provide a policy name with the -n parameter."
    usage
    return 1
  fi

  if [[ -z "$policy_document" ]]; then
    errecho "ERROR: You must provide a policy document with the -p parameter."
    usage
    return 1
  fi

  response=$(aws iam create-policy \
    --policy-name "$policy_name" \
    --policy-document "$policy_document" \
    --output text \
    --query Policy.Arn)

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-policy operation failed.\n$response"
    return 1
  fi

  echo "$response"
}

###############################################################################
# function iam_attach_role_policy
#
# This function attaches an IAM policy to a tole.
#
# 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 getopts 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
}

###############################################################################
# function iam_detach_role_policy
#
# This function detaches an IAM policy to a tole.
#
# 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 getopts 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
}

###############################################################################
# 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 getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_policy"
    echo "Deletes an WS Identity and Access Management (IAM) policy"
    echo "  -n policy_arn -- The name of the IAM policy arn."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:h" option; do
    case "${option}" in
      n) policy_arn="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
```

```
  export OPTIND=1

  if [[ -z "$policy_arn" ]]; then
    errecho "ERROR: You must provide a policy arn with the -n parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    Policy arn:  $policy_arn"
  iecho ""

  response=$(aws iam delete-policy \
    --policy-arn "$policy_arn")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-policy operation failed.\n$response"
    return 1
  fi

  iecho "delete-policy response:$response"
  iecho

  return 0
}

###############################################################################
# function iam_delete_role
#
# This function deletes an IAM role.
#
# Parameters:
#       -n role_name -- The name of the IAM role.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function iam_delete_role() {
  local role_name response
  local option OPTARG # Required to use getopts command in a function.
```

```
# bashsupport disable=BP5008
function usage() {
  echo "function iam_delete_role"
  echo "Deletes an WS Identity and Access Management (IAM) role"
  echo "  -n role_name -- The name of the IAM role."
  echo ""
}

# Retrieve the calling parameters.
while getopts "n:h" option; do
  case "${option}" in
    n) role_name="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

echo "role_name:$role_name"
if [[ -z "$role_name" ]]; then
  errecho "ERROR: You must provide a role name with the -n parameter."
  usage
  return 1
fi

iecho "Parameters:\n"
iecho "    Role name:  $role_name"
iecho ""

response=$(aws iam delete-role \
  --role-name "$role_name")

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
```

```
    errecho "ERROR: AWS reports delete-role operation failed.\n$response"
    return 1
  fi

  iecho "delete-role response:$response"
  iecho

  return 0
}

###############################################################################
# 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 getopts 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 getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_user"
    echo "Deletes an WS Identity and Access Management (IAM) user. You must supply a
 username:"
    echo "  -u user_name    The name of the user."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "u:h" option; do
    case "${option}" in
      u) user_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1
```

```
  if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    User name:   $user_name"
  iecho ""

  # If the user does not exist, we don't want to try to delete it.
  if (! iam_user_exists "$user_name"); then
    errecho "ERROR: A user with that name does not exist in the account."
    return 1
  fi

  response=$(aws iam delete-user \
    --user-name "$user_name")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-user operation failed.$response"
    return 1
  fi

  iecho "delete-user response:$response"
  iecho

  return 0
}
```

- API 자세한 내용은 AWS CLI 명령 참조의 다음 주제를 참조하세요.
  - AttachRolePolicy
  - CreateAccessKey
  - CreatePolicy
  - CreateRole
  - CreateUser
  - DeleteAccessKey

- DeletePolicy
- DeleteRole
- DeleteUser
- DeleteUserPolicy
- DetachRolePolicy
- PutUserPolicy

## 작업

### `AttachRolePolicy`

다음 코드 예시에서는 `AttachRolePolicy`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
###############################################################################
# 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 tole.
#
# 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 getopts 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
}
```

- 자세한 API 내용은 명령 참조AttachRolePolicy의 섹션을 참조하세요. AWS CLI

### `CreateAccessKey`

다음 코드 예시에서는 `CreateAccessKey`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
##############################################################################
# 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 getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_user_access_key"
    echo "Creates an AWS Identity and Access Management (IAM) key pair."
    echo "  -u user_name   The name of the IAM user."
    echo "  [-f file_name]   Optional file name for the access key output."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "u:f:h" option; do
    case "${option}" in
      u) user_name="${OPTARG}" ;;
      f) file_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
```

```
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
  fi

  response=$(aws iam create-access-key \
    --user-name "$user_name" \
    --output text)

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-access-key operation failed.$response"
    return 1
  fi

  if [[ -n "$file_name" ]]; then
    echo "$response" >"$file_name"
  fi

  local key_id key_secret
  # shellcheck disable=SC2086
  key_id=$(echo $response | cut -f 2 -d ' ')
  # shellcheck disable=SC2086
  key_secret=$(echo $response | cut -f 4 -d ' ')

  echo "$key_id $key_secret"

  return 0
}
```

- 자세한 API 내용은 명령 참조CreateAccessKey의 섹션을 참조하세요. AWS CLI

### `CreatePolicy`

다음 코드 예시에서는 `CreatePolicy`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
###############################################################################
# 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 getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_create_policy"
    echo "Creates an AWS Identity and Access Management (IAM) policy."
    echo "  -n policy_name   The name of the IAM policy."
```

```
  echo "  -p policy_json -- The policy document."
  echo ""
}

# Retrieve the calling parameters.
while getopts "n:p:h" option; do
  case "${option}" in
    n) policy_name="${OPTARG}" ;;
    p) policy_document="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$policy_name" ]]; then
  errecho "ERROR: You must provide a policy name with the -n parameter."
  usage
  return 1
fi

if [[ -z "$policy_document" ]]; then
  errecho "ERROR: You must provide a policy document with the -p parameter."
  usage
  return 1
fi

response=$(aws iam create-policy \
  --policy-name "$policy_name" \
  --policy-document "$policy_document" \
  --output text \
  --query Policy.Arn)

local error_code=${?}

if [[ $error_code -ne 0 ]]; then
  aws_cli_error_log $error_code
```

```
    errecho "ERROR: AWS reports create-policy operation failed.\n$response"
    return 1
  fi

  echo "$response"
}
```

- 자세한 API 내용은 명령 참조CreatePolicy의 섹션을 참조하세요. AWS CLI

### `CreateRole`

다음 코드 예시에서는 `CreateRole`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
#############################################################################
# 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 getopts 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
}
```

- 자세한 API 내용은 명령 참조CreateRole의 섹션을 참조하세요. AWS CLI

### `CreateUser`

다음 코드 예시에서는 `CreateUser`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
#############################################################################
# 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 getopts 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 getopts "u:h" option; do
  case "${option}" in
    u) user_name="${OPTARG}" ;;
    h)
      usage
      return 0
      ;;
    \?)
      echo "Invalid parameter"
      usage
      return 1
      ;;
  esac
done
export OPTIND=1

if [[ -z "$user_name" ]]; then
  errecho "ERROR: You must provide a username with the -u parameter."
  usage
  return 1
fi

iecho "Parameters:\n"
iecho "    User name:   $user_name"
iecho ""

# If the user 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
}
```

- 자세한 API 내용은 명령 참조CreateUser의 섹션을 참조하세요. AWS CLI

### `DeleteAccessKey`

다음 코드 예시에서는 `DeleteAccessKey`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
##############################################################################
# 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 getopts 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
}
```

- 자세한 API 내용은 명령 참조DeleteAccessKey의 섹션을 참조하세요. AWS CLI

### `DeletePolicy`

다음 코드 예시에서는 `DeletePolicy`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
#############################################################################
# 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 getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_policy"
    echo "Deletes an WS Identity and Access Management (IAM) policy"
    echo "  -n policy_arn -- The name of the IAM policy arn."
    echo ""
  }

  # Retrieve the calling parameters.
```

```
  while getopts "n:h" option; do
    case "${option}" in
      n) policy_arn="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$policy_arn" ]]; then
    errecho "ERROR: You must provide a policy arn with the -n parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    Policy arn:  $policy_arn"
  iecho ""

  response=$(aws iam delete-policy \
    --policy-arn "$policy_arn")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-policy operation failed.\n$response"
    return 1
  fi

  iecho "delete-policy response:$response"
  iecho

  return 0
}
```

- 자세한 API 내용은 명령 참조DeletePolicy의 섹션을 참조하세요. AWS CLI

### `DeleteRole`

다음 코드 예시에서는 `DeleteRole`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
###############################################################################
# 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 getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_role"
    echo "Deletes an WS Identity and Access Management (IAM) role"
    echo "  -n role_name -- The name of the IAM role."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "n:h" option; do
    case "${option}" in
      n) role_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  echo "role_name:$role_name"
  if [[ -z "$role_name" ]]; then
    errecho "ERROR: You must provide a role name with the -n parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    Role name:  $role_name"
  iecho ""
```

```
  response=$(aws iam delete-role \
    --role-name "$role_name")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-role operation failed.\n$response"
    return 1
  fi

  iecho "delete-role response:$response"
  iecho

  return 0
}
```

- 자세한 API 내용은 명령 참조DeleteRole의 섹션을 참조하세요. AWS CLI

### `DeleteUser`

다음 코드 예시에서는 `DeleteUser`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
##############################################################################
# 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 getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function iam_delete_user"
    echo "Deletes an WS Identity and Access Management (IAM) user. You must supply a
 username:"
    echo "  -u user_name    The name of the user."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "u:h" option; do
    case "${option}" in
      u) user_name="${OPTARG}" ;;
      h)
        usage
        return 0
```

```
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
  fi

  iecho "Parameters:\n"
  iecho "    User name:   $user_name"
  iecho ""

  # If the user does not exist, we don't want to try to delete it.
  if (! iam_user_exists "$user_name"); then
    errecho "ERROR: A user with that name does not exist in the account."
    return 1
  fi

  response=$(aws iam delete-user \
    --user-name "$user_name")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports delete-user operation failed.$response"
    return 1
  fi

  iecho "delete-user response:$response"
  iecho

  return 0
}
```

- 자세한 API 내용은 명령 참조DeleteUser의 섹션을 참조하세요. AWS CLI

### `DetachRolePolicy`

다음 코드 예시에서는 `DetachRolePolicy`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
###############################################################################
# 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 tole.
#
# 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 getopts 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
}
```

- 자세한 API 내용은 명령 참조DetachRolePolicy의 섹션을 참조하세요. AWS CLI

### `GetUser`

다음 코드 예시에서는 `GetUser`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
#############################################################################
# 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
}
```

- 자세한 API 내용은 명령 참조GetUser의 섹션을 참조하세요. AWS CLI

### `ListAccessKeys`

다음 코드 예시에서는 `ListAccessKeys`을 사용하는 방법을 보여 줍니다.

## AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
##############################################################################
# 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 getopts command in a function.
  # Retrieve the calling parameters.
  while getopts "u:h" option; do
    case "${option}" in
      u) user_name="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done
  export OPTIND=1

  if [[ -z "$user_name" ]]; then
    errecho "ERROR: You must provide a username with the -u parameter."
    usage
    return 1
  fi

  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
}
```

- 자세한 API 내용은 명령 참조ListAccessKeys의 섹션을 참조하세요. AWS CLI

### `ListUsers`

다음 코드 예시에서는 `ListUsers`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
##############################################################################
# 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 getopts 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 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=$(aws iam list-users \
    --output text \
    --query "Users[].UserName")
  error_code=${?}

  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
}
```

- 자세한 API 내용은 명령 참조ListUsers의 섹션을 참조하세요. AWS CLI

# Bash 스크립트 AWS CLI 와 함께 를 사용하는 Amazon S3 예제

다음 코드 예제에서는 Amazon S3에서 Bash 스크립트 AWS Command Line Interface 와 함께 를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다.

기본 사항은 서비스 내에서 필수 작업을 수행하는 방법을 보여주는 코드 예제입니다.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제



## 기본 사항

### 기본 사항 알아보기

다음 코드 예시는 다음과 같은 작업을 수행하는 방법을 보여줍니다.

- 버킷을 만들고 버킷에 파일을 업로드합니다.
- 버킷에서 객체를 다운로드합니다.
- 버킷의 하위 폴더에 객체를 복사합니다.
- 버킷의 객체를 나열합니다.
- 버킷 객체와 버킷을 삭제합니다.

### AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
###############################################################################
# function s3_getting_started
#
# This function creates, copies, and deletes S3 buckets and objects.
#
# Returns:
#       0 - If successful.
#       1 - If an error occurred.
###############################################################################
function s3_getting_started() {
  {
    if [ "$BUCKET_OPERATIONS_SOURCED" != "True" ]; then
      cd bucket-lifecycle-operations || exit

      source ./bucket_operations.sh
      cd ..
    fi
  }

  echo_repeat "*" 88
  echo "Welcome to the Amazon S3 getting started demo."
  echo_repeat "*" 88
    echo "A unique bucket will be created by appending a Universally Unique
 Identifier to a bucket name prefix."
    echo -n "Enter a prefix for the S3 bucket that will be used in this demo: "
    get_input
    bucket_name_prefix=$get_input_result
  local bucket_name
  bucket_name=$(generate_random_name "$bucket_name_prefix")

  local region_code
  region_code=$(aws configure get region)

  if create_bucket -b "$bucket_name" -r "$region_code"; then
    echo "Created demo bucket named $bucket_name"
  else
    errecho "The bucket failed to create. This demo will exit."
    return 1
  fi

  local file_name
  while [ -z "$file_name" ]; do
    echo -n "Enter a file you want to upload to your bucket: "
```

```
    get_input
    file_name=$get_input_result

    if [ ! -f "$file_name" ]; then
      echo "Could not find file $file_name. Are you sure it exists?"
      file_name=""
    fi
  done

  local key
  key="$(basename "$file_name")"

  local result=0
  if copy_file_to_bucket "$bucket_name" "$file_name" "$key"; then
    echo "Uploaded file $file_name into bucket $bucket_name with key $key."
  else
    result=1
  fi

  local destination_file
  destination_file="$file_name.download"
  if yes_no_input "Would you like to download $key to the file $destination_file?
 (y/n) "; then
    if download_object_from_bucket "$bucket_name" "$destination_file" "$key"; then
      echo "Downloaded $key in the bucket $bucket_name to the file
 $destination_file."
    else
      result=1
    fi
  fi

  if yes_no_input "Would you like to copy $key a new object key in your bucket? (y/
n) "; then
    local to_key
    to_key="demo/$key"
    if copy_item_in_bucket "$bucket_name" "$key" "$to_key"; then
      echo "Copied $key in the bucket $bucket_name to the  $to_key."
    else
      result=1
    fi
  fi

  local bucket_items
  bucket_items=$(list_items_in_bucket "$bucket_name")
```

```
  # shellcheck disable=SC2181
  if [[ $? -ne 0 ]]; then
    result=1
  fi

  echo "Your bucket contains the following items."
  echo -e "Name\t\tSize"
  echo "$bucket_items"

  if yes_no_input "Delete the bucket, $bucket_name, as well as the objects in it?
 (y/n) "; then
    bucket_items=$(echo "$bucket_items" | cut -f 1)

    if delete_items_in_bucket "$bucket_name" "$bucket_items"; then
      echo "The following items were deleted from the bucket $bucket_name"
      echo "$bucket_items"
    else
      result=1
    fi

    if delete_bucket "$bucket_name"; then
      echo "Deleted the bucket $bucket_name"
    else
      result=1
    fi
  fi

  return $result
}
```

이 시나리오에 사용된 Amazon S3 함수입니다.

```
###############################################################################
# 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 getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function create_bucket"
    echo "Creates an Amazon S3 bucket. You must supply a bucket name:"
    echo "  -b bucket_name    The name of the bucket. It must be globally unique."
    echo "  [-r region_code]    The code for an AWS Region in which the bucket is
 created."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "b:r:h" option; do
    case "${option}" in
      b) bucket_name="${OPTARG}" ;;
      r) region_code="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done

  if [[ -z "$bucket_name" ]]; then
    errecho "ERROR: You must provide a bucket name with the -b parameter."
    usage
    return 1
  fi
```

```
  local bucket_config_arg
  # A location constraint for "us-east-1" returns an error.
  if [[ -n "$region_code" ]] && [[ "$region_code" != "us-east-1" ]]; then
    bucket_config_arg="--create-bucket-configuration LocationConstraint=
$region_code"
  fi

  iecho "Parameters:\n"
  iecho "    Bucket name:   $bucket_name"
  iecho "    Region code:   $region_code"
  iecho ""

  # If the bucket already exists, we don't want to try to create it.
  if (bucket_exists "$bucket_name"); then
    errecho "ERROR: A bucket with that name already exists. Try again."
    return 1
  fi

  # shellcheck disable=SC2086
  response=$(aws s3api create-bucket \
    --bucket "$bucket_name" \
    $bucket_config_arg)

  # shellcheck disable=SC2181
  if [[ ${?} -ne 0 ]]; then
    errecho "ERROR: AWS reports create-bucket operation failed.\n$response"
    return 1
  fi
}

################################################################################
# function copy_file_to_bucket
#
# This function creates a file in the specified bucket.
#
# Parameters:
#       $1 - The name of the bucket to copy the file to.
#       $2 - The path and file name of the local file to copy to the bucket.
#       $3 - The key (name) to call the copy of the file in the bucket.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
```

```
################################################################################
function copy_file_to_bucket() {
  local response bucket_name source_file destination_file_name
  bucket_name=$1
  source_file=$2
  destination_file_name=$3

  response=$(aws s3api put-object \
    --bucket "$bucket_name" \
    --body "$source_file" \
    --key "$destination_file_name")

  # shellcheck disable=SC2181
  if [[ ${?} -ne 0 ]]; then
    errecho "ERROR: AWS reports put-object operation failed.\n$response"
    return 1
  fi
}

################################################################################
# function download_object_from_bucket
#
# This function downloads an object in a bucket to a file.
#
# Parameters:
#       $1 - The name of the bucket to download the object from.
#       $2 - The path and file name to store the downloaded bucket.
#       $3 - The key (name) of the object in the bucket.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
################################################################################
function download_object_from_bucket() {
  local bucket_name=$1
  local destination_file_name=$2
  local object_name=$3
  local response

  response=$(aws s3api get-object \
    --bucket "$bucket_name" \
    --key "$object_name" \
    "$destination_file_name")
```

```
  # shellcheck disable=SC2181
  if [[ ${?} -ne 0 ]]; then
    errecho "ERROR: AWS reports put-object operation failed.\n$response"
    return 1
  fi
}

################################################################################
# function copy_item_in_bucket
#
# This function creates a copy of the specified file in the same bucket.
#
# Parameters:
#       $1 - The name of the bucket to copy the file from and to.
#       $2 - The key of the source file to copy.
#       $3 - The key of the destination file.
#
# Returns:
#       0 - If successful.
#       1 - If it fails.
################################################################################
function copy_item_in_bucket() {
  local bucket_name=$1
  local source_key=$2
  local destination_key=$3
  local response

  response=$(aws s3api copy-object \
    --bucket "$bucket_name" \
    --copy-source "$bucket_name/$source_key" \
    --key "$destination_key")

  # shellcheck disable=SC2181
  if [[ $? -ne 0 ]]; then
    errecho "ERROR:  AWS reports s3api copy-object operation failed.\n$response"
    return 1
  fi
}

################################################################################
# function list_items_in_bucket
#
# This function displays a list of the files in the bucket with each file's
# size. The function uses the --query parameter to retrieve only the key and
```

```
# size fields from the Contents collection.
#
# Parameters:
#       $1 - The name of the bucket.
#
# Returns:
#       The list of files in text format.
#     And:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function list_items_in_bucket() {
  local bucket_name=$1
  local response

  response=$(aws s3api list-objects \
    --bucket "$bucket_name" \
    --output text \
    --query 'Contents[].{Key: Key, Size: Size}')

  # shellcheck disable=SC2181
  if [[ ${?} -eq 0 ]]; then
    echo "$response"
  else
    errecho "ERROR: AWS reports s3api list-objects operation failed.\n$response"
    return 1
  fi
}

###############################################################################
# function delete_items_in_bucket
#
# This function deletes the specified list of keys from the specified bucket.
#
# Parameters:
#       $1 - The name of the bucket.
#       $2 - A list of keys in the bucket to delete.

# Returns:
#       0 - If successful.
#       1 - If it fails.
###############################################################################
function delete_items_in_bucket() {
  local bucket_name=$1
```

```
  local keys=$2
  local response

  # Create the JSON for the items to delete.
  local delete_items
  delete_items="{\"Objects\":["
  for key in $keys; do
    delete_items="$delete_items{\"Key\": \"$key\"},"
  done
  delete_items=${delete_items%?} # Remove the final comma.
  delete_items="$delete_items]}"

  response=$(aws s3api delete-objects \
    --bucket "$bucket_name" \
    --delete "$delete_items")

  # shellcheck disable=SC2181
  if [[ $? -ne 0 ]]; then
    errecho "ERROR:  AWS reports s3api delete-object operation failed.\n$response"
    return 1
  fi
}

################################################################################
# function delete_bucket
#
# This function deletes the specified bucket.
#
# Parameters:
#       $1 - The name of the bucket.

# Returns:
#       0 - If successful.
#       1 - If it fails.
################################################################################
function delete_bucket() {
  local bucket_name=$1
  local response

  response=$(aws s3api delete-bucket \
    --bucket "$bucket_name")

  # shellcheck disable=SC2181
  if [[ $? -ne 0 ]]; then
```

```
    errecho "ERROR: AWS reports s3api delete-bucket failed.\n$response"
    return 1
  fi
}
```

- API 자세한 내용은 AWS CLI 명령 참조의 다음 주제를 참조하세요.
  - CopyObject
  - CreateBucket
  - DeleteBucket
  - DeleteObjects
  - GetObject
  - ListObjectsV2
  - PutObject

## 작업

### `CopyObject`

다음 코드 예시에서는 `CopyObject`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
##############################################################################
# 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
}
```

- 자세한 API 내용은 명령 참조CopyObject의 섹션을 참조하세요. AWS CLI

### `CreateBucket`

다음 코드 예시에서는 `CreateBucket`을 사용하는 방법을 보여 줍니다.

## AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
##############################################################################
# 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 getopts command in a function.

  # bashsupport disable=BP5008
  function usage() {
    echo "function create_bucket"
    echo "Creates an Amazon S3 bucket. You must supply a bucket name:"
    echo "  -b bucket_name    The name of the bucket. It must be globally unique."
    echo "  [-r region_code]    The code for an AWS Region in which the bucket is
 created."
    echo ""
  }

  # Retrieve the calling parameters.
  while getopts "b:r:h" option; do
    case "${option}" in
      b) bucket_name="${OPTARG}" ;;
      r) region_code="${OPTARG}" ;;
      h)
        usage
        return 0
        ;;
      \?)
        echo "Invalid parameter"
        usage
        return 1
        ;;
    esac
  done

  if [[ -z "$bucket_name" ]]; then
    errecho "ERROR: You must provide a bucket name with the -b parameter."
    usage
    return 1
  fi

  local bucket_config_arg
  # A location constraint for "us-east-1" returns an error.
  if [[ -n "$region_code" ]] && [[ "$region_code" != "us-east-1" ]]; then
```

```
    bucket_config_arg="--create-bucket-configuration LocationConstraint=
$region_code"
  fi

  iecho "Parameters:\n"
  iecho "    Bucket name:   $bucket_name"
  iecho "    Region code:   $region_code"
  iecho ""

  # If the bucket already exists, we don't want to try to create it.
  if (bucket_exists "$bucket_name"); then
    errecho "ERROR: A bucket with that name already exists. Try again."
    return 1
  fi

  # shellcheck disable=SC2086
  response=$(aws s3api create-bucket \
    --bucket "$bucket_name" \
    $bucket_config_arg)

  # shellcheck disable=SC2181
  if [[ ${?} -ne 0 ]]; then
    errecho "ERROR: AWS reports create-bucket operation failed.\n$response"
    return 1
  fi
}
```

- 자세한 API 내용은 명령 참조CreateBucket의 섹션을 참조하세요. AWS CLI

### `DeleteBucket`

다음 코드 예시에서는 `DeleteBucket`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
##############################################################################
# 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
}
```

- 자세한 API 내용은 명령 참조DeleteBucket의 섹션을 참조하세요. AWS CLI

### `DeleteObject`

다음 코드 예시에서는 `DeleteObject`을 사용하는 방법을 보여 줍니다.

## AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
##############################################################################
# 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
}
```

- 자세한 API 내용은 명령 참조DeleteObject의 섹션을 참조하세요. AWS CLI

### `DeleteObjects`

다음 코드 예시에서는 `DeleteObjects`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
##############################################################################
# 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
}
```

- 자세한 API 내용은 명령 참조DeleteObjects의 섹션을 참조하세요. AWS CLI

### `GetObject`

다음 코드 예시에서는 `GetObject`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
#############################################################################
```

```
# 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
}
```

- 자세한 API 내용은 명령 참조GetObject의 섹션을 참조하세요. AWS CLI

### `HeadBucket`

다음 코드 예시에서는 HeadBucket을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
###############################################################################
# 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
}
```

- 자세한 API 내용은 명령 참조HeadBucket의 섹션을 참조하세요. AWS CLI

### `ListObjectsV2`

다음 코드 예시에서는 `ListObjectsV2`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
##############################################################################
# function errecho
#
# This function outputs everything sent to it to STDERR (standard error output).
##############################################################################
function errecho() {
  printf "%s\n" "$*" 1>&2
}

##############################################################################
# function list_items_in_bucket
#
# This function displays a list of the files in the bucket with each file's
# size. The function uses the --query parameter to retrieve only the key and
# size fields from the Contents collection.
#
# Parameters:
#       $1 - The name of the bucket.
#
# Returns:
#       The list of files in text format.
#     And:
#       0 - If successful.
#       1 - If it fails.
##############################################################################
function list_items_in_bucket() {
  local bucket_name=$1
  local response

  response=$(aws s3api list-objects \
    --bucket "$bucket_name" \
```

```
    --output text \
    --query 'Contents[].{Key: Key, Size: Size}')

  # shellcheck disable=SC2181
  if [[ ${?} -eq 0 ]]; then
    echo "$response"
  else
    errecho "ERROR: AWS reports s3api list-objects operation failed.\n$response"
    return 1
  fi
}
```

- API 자세한 내용은 AWS CLI 명령 참조의 ListObjectsV2를 참조하세요.

### `PutObject`

다음 코드 예시에서는 `PutObject`을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
##############################################################################
# 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
}
```

- 자세한 API 내용은 명령 참조PutObject의 섹션을 참조하세요. AWS CLI

## AWS STS 를 Bash 스크립트 AWS CLI 와 함께 사용하는 예제

다음 코드 예제에서는 와 AWS Command Line Interface 함께 Bash 스크립트를 사용하여 작업을 수행하고 일반적인 시나리오를 구현하는 방법을 보여줍니다 AWS STS.

작업은 대규모 프로그램에서 발췌한 코드이며 컨텍스트에 맞춰 실행해야 합니다. 작업은 개별 서비스 함수를 직접적으로 호출하는 방법을 보여주며 관련 시나리오의 컨텍스트에 맞는 작업을 볼 수 있습니다.

각 예제에는 컨텍스트에서 코드를 설정하고 실행하는 방법에 대한 지침을 찾을 수 있는 전체 소스 코드에 대한 링크가 포함되어 있습니다.

주제

- 작업

## 작업

### `AssumeRole`

다음 코드 예시에서는 AssumeRole을 사용하는 방법을 보여 줍니다.

AWS CLI Bash 스크립트 사용

> **Note**
>
> 에 대한 자세한 내용은 를 참조하세요 GitHub. AWS 코드 예시 리포지토리에서 전체 예시를 찾고 설정 및 실행하는 방법을 배워보세요.

```
##############################################################################
# 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 getopts 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 getopts n:r:h option; do
    case "${option}" in
      n) role_session_name=${OPTARG} ;;
      r) role_arn=${OPTARG} ;;
      h)
        usage
        return 0
        ;;
      \?)
        ech o"Invalid parameter"
        usage
        return 1
        ;;
    esac
  done

  response=$(aws sts assume-role \
    --role-session-name "$role_session_name" \
    --role-arn "$role_arn" \
    --output text \
```

```
    --query "Credentials.[AccessKeyId, SecretAccessKey, SessionToken]")

  local error_code=${?}

  if [[ $error_code -ne 0 ]]; then
    aws_cli_error_log $error_code
    errecho "ERROR: AWS reports create-role operation failed.\n$response"
    return 1
  fi

  echo "$response"

  return 0
}
```

- 자세한 API 내용은 명령 참조AssumeRole의 섹션을 참조하세요. AWS CLI

# 의 보안 AWS CLI

의 클라우드 보안 AWS 이 최우선 순위입니다. AWS 고객은 가장 보안에 민감한 조직의 요구 사항을 충족하도록 구축된 데이터 센터 및 네트워크 아키텍처의 이점을 누릴 수 있습니다.

보안은 AWS 와 사용자 간의 공동 책임입니다. 공동 책임 모델 에서는 이를 클라우드 자체의 보안과 클라우드 내부의 보안으로 설명합니다.

- 클라우드 보안 - AWS 는 클라우드에서 AWS AWS 서비스를 실행하는 인프라를 보호할 책임이 있습니다. AWS 또한 는 안전하게 사용할 수 있는 서비스를 제공합니다. 타사 감사자는 AWS 규정 준수 프로그램 규정 준수 일환으로 보안의 효과를 정기적으로 테스트하고 확인합니다. 에 적용되는 규정 준수 프로그램에 대한 자세한 내용은 규정 준수 프로그램 AWS 범위 내 서비스 규정 준수 프로그램 AWS Command Line Interface참조하세요.
- 클라우드의 보안 - 사용자의 책임은 사용하는 AWS 서비스에 따라 결정됩니다. 또한 귀하는 귀사의 데이터의 민감도, 귀사의 요구 사항, 관련 법률 및 규정을 비롯한 기타 요소에 대해서도 책임이 있습니다.

이 설명서는 AWS Command Line Interface ()를 사용할 때 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 됩니다AWS CLI. 다음 주제에서는 보안 및 규정 준수 목표에 맞게 AWS CLI 를 구성하는 방법을 보여줍니다. 또한 를 사용하여 AWS 리소스를 AWS CLI 모니터링하고 보호하는 방법을 알아봅니다.

주제

- 의 데이터 보호 AWS CLI
- ID 및 액세스 관리
- 이 AWS 제품 또는 서비스에 대한 규정 준수 검증
- 이 AWS 제품 또는 서비스에 대한 복원력
- 이 AWS 제품 또는 서비스에 대한 인프라 보안
- 에 TLS 대한 최소 버전 적용 AWS CLI

# 의 데이터 보호 AWS CLI

AWS 공동 책임 모델 의 데이터 보호에 적용됩니다 AWS Command Line Interface. 이 모델에 설명된 대로 AWS 는 모든 를 실행하는 글로벌 인프라를 보호할 책임이 있습니다 AWS 클라우드. 사용자는 인

프라에서 호스팅되는 콘텐츠를 관리해야 합니다. 사용하는 AWS 서비스 의 보안 구성과 관리 작업에 대한 책임도 사용자에게 있습니다. 데이터 프라이버시에 대한 자세한 내용은 데이터 프라이버시 섹션을 FAQ참조하세요. 유럽의 데이터 보호에 대한 자세한 내용은 AWS 보안 블로그의 AWS 책임 공유 모델 및 GDPR 블로그 게시물을 참조하세요.

데이터 보호를 위해 자격 증명을 보호하고 AWS 계정 AWS IAM Identity Center 또는 AWS Identity and Access Management ()를 사용하여 개별 사용자를 설정하는 것이 좋습니다IAM. 이렇게 하면 개별 사용자에게 자신의 직무를 충실히 이행하는 데 필요한 권한만 부여됩니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정에 다단계 인증(MFA)을 사용합니다.
- SSL/TLS를 사용하여 AWS 리소스와 통신합니다. TLS 1.2가 필요하며 TLS 1.3을 권장합니다.
- 를 사용하여 API 및 사용자 활동 로깅을 설정합니다 AWS CloudTrail. CloudTrail 추적을 사용하여 AWS 활동을 캡처하는 방법에 대한 자세한 내용은 AWS CloudTrail 사용 설명서의 CloudTrail 추적 작업을 참조하세요.
- AWS 암호화 솔루션과 의 모든 기본 보안 제어를 사용합니다 AWS 서비스.
- Amazon S3에 저장된 민감한 데이터를 검색하고 보호하는 데 도움이 되는 Amazon Macie와 같은 고급 관리형 보안 서비스를 사용하세요.
- 명령줄 인터페이스 또는 FIPS 를 AWS 통해 에 액세스할 때 140-3 검증 암호화 모듈이 필요한 경우 FIPS 엔드포인트를 API사용합니다. 사용 가능한 FIPS 엔드포인트에 대한 자세한 내용은 연방 정보 처리 표준(FIPS) 140-3을 참조하세요.

고객의 이메일 주소와 같은 기밀 정보나 중요한 정보는 태그나 이름 필드와 같은 자유 양식 필드에 입력하지 않는 것이 좋습니다. 여기에는 콘솔, API AWS CLI, 또는 를 사용하여 AWS CLI 또는 다른 AWS 서비스 로 작업하는 경우가 포함됩니다 AWS SDKs. 이름에 사용되는 태그 또는 자유 형식 텍스트 필드에 입력하는 모든 데이터는 청구 또는 진단 로그에 사용될 수 있습니다. 외부 서버에 URL를 제공하는 경우 해당 서버에 대한 요청을 검증URL하기 위해 에 보안 인증 정보를 포함하지 않는 것이 좋습니다.

## 데이터 암호화

보안 서비스의 주요 특징은 정보가 활발히 사용되지 않을 때 암호화된다는 것입니다.

### 저장 중 암호화

AWS CLI 는 사용자를 대신하여 AWS 서비스와 상호 작용하는 데 필요한 자격 증명 이외의 고객 데이터를 자체적으로 저장하지 않습니다.

AWS CLI 를 사용하여 로컬 컴퓨터에 고객 데이터를 전송하여 저장하는 AWS 서비스를 호출하는 경우 해당 데이터의 저장, 보호 및 암호화 방법에 대한 자세한 내용은 해당 서비스의 사용 설명서의 보안 및 규정 준수 장을 참조하세요.

### 전송 중 암호화

기본적으로 AWS CLI 및 AWS 서비스 엔드포인트를 실행하는 클라이언트 컴퓨터에서 전송되는 모든 데이터는 HTTPS/TLS 연결을 통해 모든 데이터를 전송하여 암호화됩니다.

HTTPS/ 사용을 활성화하기 위해 아무것도 할 필요가 없습니다TLS. `--no-verify-ssl` 명령줄 옵션을 사용하여 개별 명령에 대해 명시적으로 비활성화하지 않는 한 항상 활성화됩니다.

## ID 및 액세스 관리

AWS Identity and Access Management (IAM)는 관리자가 AWS 리소스에 대한 액세스를 안전하게 제어하는 데 도움이 AWS 서비스 되는 입니다. IAM 관리자는 AWS 누가 리소스를 사용할 수 있는 인증(로그인) 및 권한 부여(권한 부여)를 받을 수 있는지 제어합니다. IAM 는 추가 비용 없이 사용할 수 AWS 서비스 있는 입니다.

주제

- 고객
- ID를 통한 인증
- 정책을 사용한 액세스 관리
- 에서 AWS 서비스 작업하는 방법 IAM
- AWS 자격 증명 및 액세스 문제 해결

### 고객

AWS Identity and Access Management (IAM) 사용 방법은 에서 수행하는 작업에 따라 다릅니다 AWS.

서비스 사용자 - AWS 서비스 를 사용하여 작업을 수행하는 경우 관리자는 필요한 자격 증명과 권한을 제공합니다. 더 많은 AWS 기능을 사용하여 작업을 수행하면 추가 권한이 필요할 수 있습니다. 액세스 권한 관리 방식을 이해하면 적절한 권한을 관리자에게 요청할 수 있습니다. 에서 기능에 액세스할 수 없는 경우 사용 중인 의 AWS 자격 증명 및 액세스 문제 해결 또는 사용 설명서를 AWS참조 AWS 서비스 하세요.

서비스 관리자 - 회사의 AWS 리소스를 담당하는 경우 에 대한 전체 액세스 권한이 있을 수 있습니다 AWS. 서비스 사용자가 액세스해야 하는 AWS 기능과 리소스를 결정하는 것은 사용자의 작업입니다.

그런 다음 IAM 관리자에게 요청을 제출하여 서비스 사용자의 권한을 변경해야 합니다. 이 페이지의 정보를 검토하여 의 기본 개념을 이해합니다IAM. 회사에서 IAM를 사용하는 방법에 대한 자세한 내용은 사용 중인 의 AWS 서비스 사용 설명서를 AWS참조하세요.

IAM 관리자 - IAM 관리자인 경우 에 대한 액세스를 관리하기 위한 정책을 작성하는 방법에 대한 세부 정보를 알고 싶을 수 있습니다 AWS. 에서 사용할 수 있는 자격 AWS 증명 기반 정책 예제를 보려면 사용 중인 의 사용 설명서를 IAM참조 AWS 서비스 하세요.

## ID를 통한 인증

인증은 자격 증명 AWS 으로 에 로그인하는 방법입니다. 로 AWS 계정 루트 사용자, IAM 사용자로 또는 IAM 역할을 수임하여 인증(에 로그인 AWS)되어야 합니다.

자격 증명 소스를 통해 제공된 자격 증명을 사용하여 에 페더레이션 자격 증명 AWS 으로 로그인할 수 있습니다. AWS IAM Identity Center (IAM Identity Center) 사용자, 회사의 Single Sign-On 인증 및 Google 또는 Facebook 자격 증명은 페더레이션 자격 증명의 예입니다. 페더레이션 자격 증명으로 로그인하면 관리자가 이전에 IAM 역할을 사용하여 자격 증명 페더레이션을 설정합니다. 페더레이션을 사용하여 AWS 에 액세스하면 간접적으로 역할을 수임하게 됩니다.

사용자 유형에 따라 AWS Management Console 또는 AWS 액세스 포털에 로그인할 수 있습니다. 에 로그인하는 방법에 대한 자세한 내용은 AWS 로그인 사용 설명서의 에 로그인하는 방법을 AWS 계정 AWS참조하세요.

AWS 프로그래밍 방식으로 에 액세스하는 경우 는 소프트웨어 개발 키트(SDK)와 명령줄 인터페이스(CLI)를 AWS 제공하여 자격 증명을 사용하여 요청에 암호화 방식으로 서명합니다. AWS 도구를 사용하지 않는 경우 직접 요청에 서명해야 합니다. 권장 방법을 사용하여 직접 요청에 서명하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 AWS API 요청에 대한 서명 버전 4를 참조하세요.

사용하는 인증 방법에 상관없이 추가 보안 정보를 제공해야 할 수도 있습니다. 예를 들어 다중 인증(MFA)을 사용하여 계정의 보안을 강화하는 것이 AWS 좋습니다. 자세한 내용은 AWS IAM Identity Center 사용 설명서의 다단계 인증 및 사용 설명서의 AWS 다단계 인증을 IAM 참조하세요IAM.

### AWS 계정 루트 사용자

를 생성하면 계정의 모든 AWS 서비스 및 리소스에 대한 완전한 액세스 권한이 있는 로그인 자격 증명 하나로 AWS 계정시작합니다. 이 자격 증명을 AWS 계정 루트 사용자라고 하며 계정을 생성하는 데 사용한 이메일 주소와 암호로 로그인하여 액세스합니다. 일상적인 작업에 루트 사용자를 사용하지 않을 것을 강력히 권장합니다. 루트 사용자 보안 인증 정보를 보호하고 루트 사용자만 수행할 수 있는 작업을 수행하는 데 사용합니다. 루트 사용자로 로그인해야 하는 작업의 전체 목록은 IAM 사용 설명서의 루트 사용자 보안 인증이 필요한 작업을 참조하세요.

## 페더레이션 자격 증명

가장 좋은 방법은 관리자 액세스가 필요한 사용자를 포함한 인간 사용자가 ID 공급자와의 페더레이션을 사용하여 임시 자격 증명을 사용하여 AWS 서비스 에 액세스하도록 요구하는 것입니다.

페더레이션 자격 증명은 엔터프라이즈 사용자 디렉터리, 웹 자격 증명 공급자, , AWS Directory Service Identity Center 디렉터리 또는 자격 증명 소스를 통해 제공된 자격 증명을 사용하여 AWS 서비스 에 액세스하는 모든 사용자의 사용자입니다.  페더레이션 자격 증명이 에 액세스하면 역할을 AWS 계정수임하고 역할은 임시 자격 증명을 제공합니다.

중앙 집중식 액세스 관리를 위해 AWS IAM Identity Center(을)를 사용하는 것이 좋습니다. IAM Identity Center에서 사용자 및 그룹을 생성하거나 및 애플리케이션에서 사용할 수 있도록 자체 자격 증명 소스의 사용자 AWS 계정 및 그룹 집합에 연결하고 동기화할 수 있습니다. IAM Identity Center에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서의 IAM Identity Center란 무엇입니까?를 참조하세요.

## IAM 사용자 및 그룹

IAM 사용자는 한 사람 또는 애플리케이션에 대한 특정 권한이 AWS 계정 있는 내 자격 증명입니다. 가능한 경우 암호 및 액세스 키와 같은 장기 보안 인증 정보가 있는 IAM 사용자를 생성하는 대신 임시 보안 인증 정보를 사용하는 것이 좋습니다. 그러나 IAM 사용자와 장기 보안 인증이 필요한 특정 사용 사례가 있는 경우 액세스 키를 교체하는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 장기 보안 인증이 필요한 사용 사례에 대한 액세스 키 정기적으로 교체를 참조하세요.

IAM 그룹은 IAM 사용자 컬렉션을 지정하는 자격 증명입니다. 사용자는 그룹으로 로그인할 수 없습니다. 그룹을 사용하여 여러 사용자의 권한을 한 번에 지정할 수 있습니다. 그룹을 사용하면 대규모 사용자 집합의 권한을 더 쉽게 관리할 수 있습니다. 예를 들어 라는 이름의 그룹을 지정IAMAdmins하고 해당 그룹에 IAM 리소스를 관리할 수 있는 권한을 부여할 수 있습니다.

사용자는 역할과 다릅니다. 사용자는 한 사람 또는 애플리케이션과 고유하게 연결되지만, 역할은 해당 역할이 필요한 사람이라면 누구나 수임할 수 있습니다. 사용자는 영구적인 장기 보안 인증 정보를 가지고 있지만, 역할은 임시 보안 인증만 제공합니다. 자세한 내용은 IAM 사용 설명서의 IAM 사용자 사용 사례를 참조하세요.

## IAM 역할

IAM 역할은 특정 권한이 AWS 계정 있는 내 자격 증명입니다. IAM 사용자와 비슷하지만 특정 사람과는 연결되지 않습니다. 에서 IAM 역할을 일시적으로 수임하려면 사용자에서 역할(콘솔)로 전환할 AWS Management Console수 있습니다. IAM  또는 AWS API 작업을 호출 AWS CLI 하거나 사용자 지정 를 사용하여 역할을 수임할 수 있습니다URL. 역할 사용 방법에 대한 자세한 내용은 IAM 사용 설명서의 역할 수임 방법을 참조하세요.

IAM 임시 자격 증명이 있는 역할은 다음과 같은 상황에서 유용합니다.

- 페더레이션 사용자 액세스 - 페더레이션 ID에 권한을 부여하려면 역할을 생성하고 해당 역할의 권한을 정의합니다. 페더레이션 ID가 인증되면 역할이 연결되고 역할에 정의된 권한이 부여됩니다. 페더레이션 역할에 대한 자세한 내용은 IAM 사용 설명서의 타사 자격 증명 공급자에 대한 역할 생성을 참조하세요. IAM Identity Center를 사용하는 경우 권한 세트를 구성합니다. 인증 후 자격 증명이 액세스할 수 있는 항목을 제어하기 위해 IAM Identity Center는 권한 세트를 의 역할과 상호 연관시킵니다IAM. 권한 세트에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서의 권한 세트를 참조하세요.
- 임시 IAM 사용자 권한 - IAM 사용자 또는 역할은 특정 작업에 대해 일시적으로 다른 권한을 맡을 IAM 수 있습니다.
- 교차 계정 액세스 - IAM 역할을 사용하여 다른 계정의 누군가(신뢰할 수 있는 보안 주체)가 계정의 리소스에 액세스하도록 허용할 수 있습니다. 역할은 계정 간 액세스를 부여하는 기본적인 방법입니다. 그러나 일부 에서는 정책을 리소스에 직접 연결할 AWS 서비스수 있습니다(역할을 프록시로 사용하는 대신). 크로스 계정 액세스에 대한 역할과 리소스 기반 정책의 차이점을 알아보려면 IAM 사용 설명서의 에서 크로스 계정 리소스 액세스를 IAM 참조하세요.
- 교차 서비스 액세스 - 일부 는 다른 에서 기능을 AWS 서비스 사용합니다 AWS 서비스. 예를 들어 서비스에서 호출할 때 해당 서비스가 Amazon에서 애플리케이션을 실행EC2하거나 Amazon S3에 객체를 저장하는 것이 일반적입니다. 서비스는직접적으로 호출하는 보안 주체의 권한을 사용하거나, 서비스 역할을 사용하거나, 또는 서비스 연결 역할을 사용하여 이 작업을 수행할 수 있습니다.
  - 전달 액세스 세션(FAS) - IAM 사용자 또는 역할을 사용하여 에서 작업을 수행하면 보안 주체로 AWS간주됩니다. 일부 서비스를 사용하는 경우 다른 서비스에서 다른 작업을 시작하는 작업을 수행할 수 있습니다. FAS 는 를 호출하는 보안 주체의 권한을 다운스트림 서비스에 AWS 서비스 대한 요청과 AWS 서비스함께 사용합니다. FAS 요청은 서비스가 다른 AWS 서비스 또는 리소스와의 상호 작용을 완료해야 하는 요청을 수신할 때만 수행됩니다. 이 경우 두 작업을 모두 수행할 수 있는 권한이 있어야 합니다. FAS 요청 시 정책 세부 정보는 액세스 세션 전달을 참조하세요.
  - 서비스 역할 - 서비스 역할은 서비스가 사용자를 대신하여 작업을 수행하기 위해 수임하는 IAM 역할입니다. IAM 관리자는 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다IAM. 자세한 내용은 IAM 사용 설명서의 에 권한을 위임할 역할 생성을 AWS 서비스 참조하세요.
  - 서비스 연결 역할 - 서비스 연결 역할은 에 연결된 서비스 역할의 한 유형입니다 AWS 서비스. 서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은 에 표시 AWS 계정 되며 서비스가 소유합니다. IAM 관리자는 서비스 연결 역할에 대한 권한을 볼 수 있지만 편집할 수는 없습니다.
- Amazon에서 실행되는 애플리케이션 EC2 - IAM 역할을 사용하여 EC2 인스턴스에서 실행되고 AWS CLI 또는 AWS API 요청을 수행하는 애플리케이션의 임시 보안 인증을 관리할 수 있습니다. 이는

EC2 인스턴스 내에 액세스 키를 저장하는 것보다 좋습니다. EC2 인스턴스에 AWS 역할을 할당하고 모든 애플리케이션에서 사용할 수 있도록 하려면 인스턴스에 연결된 인스턴스 프로파일을 생성합니다. 인스턴스 프로파일에는 역할이 포함되어 있으며 EC2 인스턴스에서 실행 중인 프로그램이 임시 자격 증명을 가져올 수 있습니다. 자세한 내용은 IAM 사용 설명서EC2의 IAM 역할 사용을 참조하세요.

## 정책을 사용한 액세스 관리

정책을 AWS 생성하고 AWS 자격 증명 또는 리소스에 연결하여 의 액세스를 제어합니다. 정책은 자격 증명 또는 리소스와 연결된 AWS 경우 권한을 정의하는 의 객체입니다. 는 보안 주체(사용자, 루트 사용자 또는 역할 세션)가 요청할 때 이러한 정책을 AWS 평가합니다. 정책에서 권한은 요청이 허용되거나 거부되는 지를 결정합니다. 대부분의 정책은 에 JSON 문서 AWS 로 저장됩니다. JSON 정책 문서의 구조 및 내용에 대한 자세한 내용은 IAM 사용 설명서의 JSON 정책 개요를 참조하세요.

관리자는 정책을 사용하여 AWS JSON 대상에 액세스할 수 있는 사용자를 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

기본적으로, 사용자와 역할에는 어떠한 권한도 없습니다. 사용자에게 필요한 리소스에 대한 작업을 수행할 수 있는 권한을 부여하기 위해 IAM 관리자는 IAM 정책을 생성할 수 있습니다. 그런 다음 관리자는 IAM 정책을 역할에 추가하고 사용자는 역할을 수임할 수 있습니다.

IAM 정책은 작업을 수행하는 데 사용하는 방법에 관계없이 작업에 대한 권한을 정의합니다. 예를 들어, `iam:GetRole` 작업을 허용하는 정책이 있다고 가정합니다. 해당 정책을 사용하는 사용자는 AWS Management Console, AWS CLI또는 에서 역할 정보를 가져올 수 있습니다 AWS API.

### 보안 인증 기반 정책

자격 증명 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 자격 증명에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자와 역할이 어떤 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 제어합니다. 자격 증명 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 고객 관리형 정책을 사용하여 사용자 지정 IAM 권한 정의를 참조하세요.

보안 인증 기반 정책은 인라인 정책 또는 관리형 정책으로 한층 더 분류할 수 있습니다. 인라인 정책은 단일 사용자, 그룹 또는 역할에 직접 포함됩니다. 관리형 정책은 의 여러 사용자, 그룹 및 역할에 연결할 수 있는 독립 실행형 정책입니다 AWS 계정. 관리형 정책에는 AWS 관리형 정책 및 고객 관리형 정책이 포함됩니다. 관리형 정책 또는 인라인 정책 중에서 선택하는 방법을 알아보려면 IAM 사용 설명서의 관리형 정책 및 인라인 정책 중에서 선택을 참조하세요.

## 리소스 기반 정책

리소스 기반 정책은 리소스에 연결하는 JSON 정책 문서입니다. 리소스 기반 정책의 예로는 IAM 역할 신뢰 정책 및 Amazon S3 버킷 정책이 있습니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는 이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 정책이 연결된 리소스의 경우 정책은 지정된 보안 주체가 해당 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는지를 정의합니다. 리소스 기반 정책에서 보안 주체를 지정해야 합니다. 보안 주체에는 계정, 사용자, 역할, 페더레이션 사용자 또는 가 포함될 수 있습니다 AWS 서비스.

리소스 기반 정책은 해당 서비스에 있는 인라인 정책입니다. 리소스 기반 정책IAM에서는 의 AWS 관리형 정책을 사용할 수 없습니다.

## 액세스 제어 목록(ACLs)

액세스 제어 목록(ACLs)은 리소스에 액세스할 수 있는 권한이 있는 보안 주체(계정 멤버, 사용자 또는 역할)를 제어합니다. ACLs 는 리소스 기반 정책과 유사하지만 JSON 정책 문서 형식을 사용하지는 않습니다.

Amazon S3 AWS WAF및 AmazonVPC은 를 지원하는 서비스의 예입니다ACLs. 에 대한 자세한 내용은 Amazon Simple Storage Service 개발자 안내서의 액세스 제어 목록(ACL) 개요를 ACLs참조하세요.

## 기타 정책 타입

AWS 는 덜 일반적인 추가 정책 유형을 지원합니다. 이러한 정책 타입은 더 일반적인 정책 유형에 따라 사용자에게 부여되는 최대 권한을 설정할 수 있습니다.

- 권한 경계 - 권한 경계는 자격 증명 기반 정책이 IAM엔터티(IAM 사용자 또는 역할)에 부여할 수 있는 최대 권한을 설정하는 고급 기능입니다. 개체에 대한 권한 경계를 설정할 수 있습니다. 그 결과로 얻는 권한은 객체의 자격 증명 기반 정책과 그 권한 경계의 교집합입니다. `Principal` 필드에서 사용자나 역할을 지정하는 리소스 기반 정책은 권한 경계를 통해 제한되지 않습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 권한 경계에 대한 자세한 내용은 IAM 사용 설명서의 IAM엔터티에 대한 권한 경계를 참조하세요.
- 서비스 제어 정책(SCPs) - 의 조직 또는 조직 단위(OU)에 대한 최대 권한을 지정하는 SCPs JSON 정책입니다 AWS Organizations. AWS Organizations 는 비즈니스가 소유 AWS 계정 한 여러 을 그룹화하고 중앙에서 관리하기 위한 서비스입니다. 조직의 모든 기능을 활성화하면 서비스 제어 정책(SCPs)을 계정의 일부 또는 전체에 적용할 수 있습니다. 는 각 를 포함하여 멤버 계정의 엔터티에 대한 권한을 SCP 제한합니다 AWS 계정 루트 사용자. 조직 및 에 대한 자세한 내용은 AWS Organizations 사용 설명서의 서비스 제어 정책을 SCPs참조하세요.

- 세션 정책 – 세션 정책은 역할 또는 페더레이션 사용자에 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 고급 정책입니다. 결과적으로 얻는 세션의 권한은 사용자 또는 역할의 보안 인증 기반 정책의 교차와 세션 정책입니다. 또한 권한을 리소스 기반 정책에서 가져올 수도 있습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 자세한 내용은 IAM 사용 설명서의 세션 정책을 참조하세요.

### 여러 정책 유형

여러 정책 유형이 요청에 적용되는 경우, 결과 권한은 이해하기가 더 복잡합니다. AWS 에서 여러 정책 유형이 관련될 때 요청을 허용할지 여부를 결정하는 방법을 알아보려면 IAM 사용 설명서의 정책 평가 로직을 참조하세요.

## 에서 AWS 서비스 작업하는 방법 IAM

대부분의 IAM 기능을 AWS 서비스 사용하는 방법을 자세히 알아보려면 IAM 사용 설명서의 에서 AWS 를 사용하는 서비스를 IAM 참조하세요.

AWS 서비스 에서 특정 를 사용하는 방법을 알아보려면 관련 서비스 사용 설명서의 보안 섹션을 IAM참조하세요.

## AWS 자격 증명 및 액세스 문제 해결

다음 정보를 사용하여 AWS 및 작업 시 발생할 수 있는 일반적인 문제를 진단하고 해결할 수 있습니다 IAM.

주제

- 에서 작업을 수행할 권한이 없습니다. AWS
- iam을 수행할 권한이 없습니다.PassRole
- 내 외부의 사람들이 내 AWS 리소스에 액세스 AWS 계정 하도록 허용하고 싶습니다.

### 에서 작업을 수행할 권한이 없습니다. AWS

작업을 수행할 수 있는 권한이 없다는 오류가 수신되면 작업을 수행할 수 있도록 정책을 업데이트해야 합니다.

다음 예제 오류는 `mateojackson` IAM 사용자가 콘솔을 사용하여 가상 *`my-example-widget`* 리소스에 대한 세부 정보를 보려고 하지만 가상 `awes:`*`GetWidget`* 권한이 없는 경우에 발생합니다.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 awes:GetWidget on resource: my-example-widget
```

이 경우 awes:*GetWidget*작업을 사용하여 *my-example-widget*리소스에 액세스할 수 있도록 mateojackson사용자 정책을 업데이트해야 합니다.

도움이 필요한 경우 AWS 관리자에게 문의하세요. 관리자는 로그인 자격 증명을 제공한 사람입니다.

### iam을 수행할 권한이 없습니다.PassRole

iam:PassRole 작업을 수행할 수 있는 권한이 없다는 오류가 수신되면 AWS에 역할을 전달할 수 있도록 정책을 업데이트해야 합니다.

일부는 새 서비스 역할 또는 서비스 연결 역할을 생성하는 대신 기존 역할을 해당 서비스에 전달할 수 있도록 AWS 서비스 허용합니다. 이렇게 하려면 사용자가 서비스에 역할을 전달할 수 있는 권한을 가지고 있어야 합니다.

다음 예제 오류는 라는 IAM 사용자가 콘솔을 사용하여 에서 작업을 수행하려고 marymajor 할 때 발생합니다 AWS. 하지만 작업을 수행하려면 서비스 역할이 부여한 권한이 서비스에 있어야 합니다. Mary는 서비스에 역할을 전달할 수 있는 권한을 가지고 있지 않습니다.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

이 경우, Mary가 iam:PassRole 작업을 수행할 수 있도록 Mary의 정책을 업데이트해야 합니다.

도움이 필요한 경우 AWS 관리자에게 문의하세요. 관리자는 로그인 자격 증명을 제공한 사람입니다.

### 내 외부의 사람들이 내 AWS 리소스에 액세스 AWS 계정 하도록 허용하고 싶습니다.

다른 계정의 사용자 또는 조직 외부의 사람이 리소스에 액세스할 때 사용할 수 있는 역할을 생성할 수 있습니다. 역할을 수임할 신뢰할 수 있는 사람을 지정할 수 있습니다. 리소스 기반 정책 또는 액세스 제어 목록(ACLs)을 지원하는 서비스의 경우 이러한 정책을 사용하여 사용자에게 리소스에 대한 액세스 권한을 부여할 수 있습니다.

자세히 알아보려면 다음을 참조하세요.

- 가 이러한 기능을 AWS 지원하는지 알아보려면 섹션을 참조하세요에서 AWS 서비스 작업하는 방법 IAM.

- 소유 AWS 계정 한 의 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 소유 AWS 계정 한 다른 의 IAM 사용자에게 액세스 권한 제공을 참조하세요.
- 타사에 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 타사 AWS 계정 소유 에 대한 액세스 권한 제공을 AWS 계정참조하세요.
- 자격 증명 페더레이션을 통해 액세스를 제공하는 방법을 알아보려면 IAM 사용 설명서의 외부에서 인증된 사용자에게 액세스 제공(자격 증명 페더레이션)을 참조하세요.
- 교차 계정 액세스를 위한 역할 및 리소스 기반 정책 사용의 차이점을 알아보려면 IAM 사용 설명서의 에서 교차 계정 리소스 액세스를 IAM 참조하세요.

# 이 AWS 제품 또는 서비스에 대한 규정 준수 검증

AWS 서비스 가 특정 규정 준수 프로그램의 범위 내에 있는지 알아보려면 AWS 서비스 규정 준수 프로그램 범위 참조하고 관심 있는 규정 준수 프로그램을 선택합니다. 일반 정보는 AWS 규정 준수 프로그램 참조하세요.

를 사용하여 타사 감사 보고서를 다운로드할 수 있습니다 AWS Artifact. 자세한 내용은 의 보고서 다운로드 AWS Artifact.

를 사용할 때의 규정 준수 책임은 데이터의 민감도, 회사의 규정 준수 목표 및 관련 법률 및 규정에 따라 AWS 서비스 결정됩니다. 는 규정 준수를 지원하는 다음 리소스를 AWS 제공합니다.

- 보안 및 규정 준수 빠른 시작 안내서 - 이 배포 안내서에서는 아키텍처 고려 사항에 대해 설명하고 보안 및 규정 준수에 중점을 둔 에 기준 환경을 배포 AWS 하기 위한 단계를 제공합니다.
- Amazon Web Services의 HIPAA 보안 및 규정 준수를 위한 아키텍처 HIPAA- 이 백서에서는 기업이 AWS 를 사용하여 적격 애플리케이션을 생성하는 방법을 설명합니다.

  > **Note**
  >
  > 모두 HIPAA 적합한 AWS 서비스 것은 아닙니다. 자세한 내용은 HIPAA 적격 서비스 참조를 참조하세요.

- AWS 규정 준수 리소스 - 이 워크북 및 가이드 모음은 해당 산업 및 위치에 적용될 수 있습니다.
- AWS 고객 규정 준수 가이드 - 규정 준수의 관점에서 공동 책임 모델을 이해합니다. 이 가이드는 여러 프레임워크(미국 국립표준기술연구소(), 결제카드 산업보안표준위원회(NIST), PCI국제표준화기구(ISO) 포함)의 보안 제어에 대한 지침을 보호하고 AWS 서비스 매핑하는 모범 사례를 요약합니다.
- AWS Config 개발자 안내서의 규칙을 사용하여 리소스 평가 - 이 AWS Config 서비스는 리소스 구성이 내부 관행, 업계 지침 및 규정을 얼마나 잘 준수하는지 평가합니다.

- AWS Security Hub - AWS 서비스 에서 보안 상태를 포괄적으로 볼 수 있습니다 AWS. Security Hub 는 보안 제어를 사용하여 AWS 리소스를 평가하고 보안 업계 표준 및 모범 사례에 대한 규정 준수를 확인합니다. 지원되는 서비스 및 제어 목록은 Security Hub 제어 참조를 참조하세요.
- Amazon GuardDuty – 의심스러운 활동 및 악의적인 활동이 있는지 환경을 모니터링하여 AWS 계정, 워크로드, 컨테이너 및 데이터에 대한 잠재적 위협을 AWS 서비스 탐지합니다. 는 특정 규정 준수 프레임워크에서 요구하는 침입 탐지 요구 사항을 충족DSS하여 PCI 와 같은 다양한 규정 준수 요구 사항을 해결하는 데 도움이 될 GuardDuty 수 있습니다.
- AWS Audit Manager - 이를 AWS 서비스 통해 AWS 사용량을 지속적으로 감사하여 규정 및 업계 표준 준수 및 위험을 관리하는 방법을 간소화할 수 있습니다.

이 AWS 제품 또는 서비스는 지원하는 특정 Amazon Web Services(AWS) 서비스를 통해 공동 책임 모델을 따릅니다. AWS 서비스 보안 정보는 AWS 서비스 보안 설명서 페이지 및 규정 AWSAWS 준수 프로그램 의 규정 준수 노력 범위에 속하는 서비스를 참조하세요.

## 이 AWS 제품 또는 서비스에 대한 복원력

AWS 글로벌 인프라는 AWS 리전 및 가용 영역을 기반으로 구축됩니다.

AWS 리전 는 지연 시간이 짧고 처리량이 높으며 중복성이 높은 네트워킹과 연결된 물리적으로 분리되고 격리된 여러 가용 영역을 제공합니다.

가용 영역을 사용하면 중단 없이 영역 간에 자동으로 장애 극복 조치가 이루어지는 애플리케이션 및 데이터베이스를 설계하고 운영할 수 있습니다. 가용 영역은 기존의 단일 또는 다중 데이터 센터 인프라보다 가용성, 내결함성, 확장성이 뛰어납니다.

AWS 리전 및 가용 영역에 대한 자세한 내용은 AWS 글로벌 인프라 섹션을 참조하세요.

이 AWS 제품 또는 서비스는 지원하는 특정 Amazon Web Services(AWS) 서비스를 통해 공동 책임 모델을 따릅니다. AWS 서비스 보안 정보는 AWS 서비스 보안 설명서 페이지 및 규정 AWSAWS 준수 프로그램 의 규정 준수 노력 범위에 속하는 서비스를 참조하세요.

## 이 AWS 제품 또는 서비스에 대한 인프라 보안

이 AWS 제품 또는 서비스는 관리형 서비스를 사용하므로 글로벌 네트워크 보안으로 AWS 보호됩니다. AWS 보안 서비스 및 인프라 AWS 보호 방법에 대한 자세한 내용은 AWS 클라우드 보안 섹션을 참조하세요. 인프라 보안 모범 사례를 사용하여 AWS 환경을 설계하려면 Security Pillar AWS Well-Architected Framework의 인프라 보호를 참조하세요.

AWS 게시된 API 호출을 사용하여 네트워크를 통해 이 AWS 제품 또는 서비스에 액세스합니다. 고객은 다음을 지원해야 합니다.

- 전송 계층 보안(TLS). TLS 1.2가 필요하며 TLS 1.3을 권장합니다.
- DHE (Ephemeral Diffie-HellmanPFS) 또는 (Elliptic Curve Ephemeral Diffie-Hellman)과 같은 완벽한 순방향 보안ECDHE()이 포함된 Cipher 제품군입니다. Java 7 이상의 최신 시스템은 대부분 이러한 모드를 지원합니다.

또한 액세스 키 ID와 IAM 보안 주체와 연결된 보안 액세스 키를 사용하여 요청에 서명해야 합니다. 또는 AWS Security Token Service(AWS STS)를 사용하여 임시 보안 인증을 생성하여 요청에 서명할 수 있습니다.

이 AWS 제품 또는 서비스는 지원하는 특정 Amazon Web Services(AWS) 서비스를 통해 공동 책임 모델을 따릅니다. AWS 서비스 보안 정보는 AWS 서비스 보안 설명서 페이지 및 규정 AWSAWS 준수 프로그램 의 규정 준수 노력 범위에 속하는 서비스를 참조하세요.

# 에 TLS 대한 최소 버전 적용 AWS CLI

AWS Command Line Interface (AWS CLI)를 사용하는 경우 전송 계층 보안(TLS) 프로토콜은 AWS CLI 및 간의 통신을 보호하는 데 중요한 역할을 합니다 AWS 서비스. AWS 서비스와 통신할 때 보안을 강화하려면 TLS 1.2 이상을 사용해야 합니다.

AWS CLI 및 는 암호화, 인증 및 데이터 무결성을 제공하는 TLS 프로토콜로 데이터를 안전하게 교환할 AWS 서비스 수 있습니다. TLS 프로토콜을 활용하면 AWS CLI 와의 상호 작용 AWS 서비스 이 무단 액세스 및 데이터 침해로부터 보호되어 AWS 에코시스템의 전반적인 보안이 강화됩니다.

AWS 공동 책임 모델 의 데이터 보호에 적용됩니다 AWS Command Line Interface. 이 모델에 설명된 대로 AWS 는 모든 를 실행하는 글로벌 인프라를 보호할 책임이 있습니다 AWS 서비스. 사용자는 인프라에서 호스팅되는 콘텐츠를 관리해야 합니다. 또한 사용하는 의 보안 구성 및 관리 태스크에 대한 책임이 AWS 서비스 있습니다. 데이터 보호에 대한 자세한 내용은 섹션을 참조하세요the section called “데이터 보호”.

AWS CLI 버전 1이 TLS 1.2 이전 TLS 버전을 사용하지 않도록 하려면 OpenSSL을 다시 컴파일하여 이 최소값을 적용한 다음 Python을 다시 컴파일하여 새로 빌드된 Open 을 사용해야 할 수 있습니다SSL.

주제

- 현재 지원되는 프로토콜 확인
- OpenSSL 및 Python 컴파일

## 현재 지원되는 프로토콜 확인

먼저 Open 을 SDK 사용하여 테스트 서버 및 Python에 사용할 자체 서명된 인증서를 생성합니다SSL.

```
$ openssl req -subj '/CN=localhost' -x509 -newkey rsa:4096 -nodes -keyout key.pem -out
 cert.pem -days 365
```

그런 다음 열기 를 사용하여 테스트 서버를 스핀업합니다SSL.

```
$ openssl s_server -key key.pem -cert cert.pem -www
```

새 터미널 창에서 가상 환경을 생성하고 PythonSDK용 를 설치합니다.

```
$ python3 -m venv test-env
source test-env/bin/activate
pip install botocore
```

SDK의 기본 HTTP 라이브러리를 `check.py` 사용하는 라는 새 Python 스크립트를 생성합니다.

```
$ 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'))
```

새 스크립트를 실행합니다.

```
$ python check.py
```

그러면 연결에 대한 세부 정보가 표시됩니다. 출력에서 "프로토콜 : "을 검색합니다. 출력이 “TLSv1.2” 이상인 경우 는 SDK 기본적으로 TLS v1.2 이상으로 설정됩니다. 이전 버전인 경우 OpenSSL 및 Python을 다시 컴파일해야 합니다.

그러나 Python 설치가 TLS v1.2 이상으로 기본 설정되어 있더라도 서버가 TLS v1.2 이상을 지원하지 않는 경우 Python이 TLS v1.2 이전 버전으로 재협상할 수 있습니다. Python이 이전 버전으로 자동으로 다시 협상하지 않는지 확인하려면 다음과 같이 테스트 서버를 다시 시작하세요.

```
$ openssl s_server -key key.pem -cert cert.pem -no_tls1_3 -no_tls1_2 -www
```

이전 버전의 Open 를 사용하는 경우 `-no_tls_3` 플래그를 사용할 수 없을 SSL수 있습니다. 이 경우 사용 중인 OpenSSL 버전이 TLS v1.3을 지원하지 않으므로 플래그를 제거합니다. 그런 다음 Python 스크립트를 다시 실행합니다.

```
$ python check.py
```

Python 설치가 TLS 1.2 이전 버전에 대해 올바르게 재협상되지 않으면 SSL 오류가 발생합니다.

```
$ 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)')))
```

연결할 수 있는 경우 OpenSSL 및 Python을 다시 컴파일하여 TLS v1.2 이전의 프로토콜 협상을 비활성화해야 합니다.

## OpenSSL 및 Python 컴파일

SDK 또는 가 TLS1.2 이전 버전에 대해 협상 AWS CLI 하지 않도록 하려면 OpenSSL 및 Python을 다시 컴파일해야 합니다. 이렇게 하려면 다음 내용을 복사하여 스크립트를 만들고 실행합니다.

```
#!/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
```

이렇게 하면 1.2보다 일찍 자동으로 협상되지 않는 정적 연결 OpenSSL이 있는 Python 버전이 TLS 컴파일됩니다. 또한 디렉터리에 OpenSSL을 설치하고 `/opt/openssl-with-min-tls1_2` 디렉터리에 Python을 설치합니다`/opt/python-with-min-tls1_2`. 이 스크립트를 실행한 후 새 버전의 Python 설치를 확인하세요.

```
$ /opt/python-with-min-tls1_2/bin/python3 --version
```

다음 사항이 인쇄되어야 합니다.

```
$ Python 3.8.1
```

이 새 버전의 Python이 TLS1.2 이전 버전을 협상하지 않는지 확인하려면 새로 설치된 Python 버전(즉, `/opt/python-with-min-tls1_2/bin/python3`)을 현재 지원되는 프로토콜 확인 사용하여 단계를 다시 실행합니다.

# 에 대한 오류 해결 AWS CLI

이 섹션에서는 문제를 해결하기 위해 따라야 할 일반적인 오류와 문제 해결 단계를 다룹니다. 먼저 일반 문제 해결을 따르는 것이 좋습니다.

목차



## 먼저 시도해야 할 일반적인 문제 해결

오류가 발생하거나 에 문제가 발생하는 경우 AWS CLI문제를 해결하는 데 도움이 되는 다음과 같은 일반적인 팁을 권장합니다.

맨 위로 이동

## AWS CLI 명령 형식 확인

명령이 존재하지 않는다는 오류가 발생하거나 명령이 설명서에서 사용 가능하다고 나열된 파라미터(`Parameter validation failed`)를 인식하지 못하는 오류가 발생할 경우 명령 형식이 잘못되었을 수 있습니다. 다음을 확인하는 것이 좋습니다.

- 명령에서 맞춤법 및 형식 오류가 있는지 확인합니다.
- 명령에서 해당 터미널에 적용되는 모든 따옴표와 이스케이프가 올바른지 확인합니다.
- AWS CLI 스켈레톤을 생성하여 명령 구조를 확인합니다.
- 의 경우 JSON 값에 대한 추가 문제 해결을 JSON참조하세요. 터미널 처리 JSON 형식에 문제가 있는 경우 Blobs를 사용하여 에 직접 JSON 데이터를 전달하여 터미널의 견적 규칙을 건너뛰는 것이 AWS CLI 좋습니다.

특정 명령을 구성하는 방법에 대한 자세한 내용은 AWS CLI 참조 가이드 를 참조하세요.

맨 위로 이동

## AWS CLI 명령이 사용 중인 AWS 리전 지 확인

> **Note**
>
> 를 사용할 AWS 리전 때 명시적으로 AWS CLI또는 기본 리전을 설정하여 를 지정해야 합니다. 지정할 수 AWS 리전 있는 모든 의 목록은 의 AWS 리전 및 엔드포인트를 참조하세요Amazon Web Services 일반 참조. 에서 사용하는 AWS 리전 지정자는 및 서비스 엔드포인트에 AWS Management Console URLs 표시되는 것과 동일한 이름 AWS CLI 입니다.

지정된 에 를 사용할 수 AWS 서비스 없거나 리소스가 다른 에 있는 경우 오류 AWS 리전 또는 예상치 못한 결과가 발생할 수 있습니다 AWS 리전. 우선 순위에 따라 AWS 리전 는 다음과 같은 방식으로 설정됩니다.

- `--region` 명령줄 옵션
- `AWS_DEFAULT_REGION` 환경 변수.
- `region` 프로파일 설정입니다.

리소스에 AWS 리전 올바른 를 사용하고 있는지 확인합니다.

맨 위로 이동

## 최신 버전의 AWS CLI를 실행 중인지 확인합니다.

명령이 존재하지 않거나 AWS CLI 참조 가이드 에서 사용할 수 있다고 말하는 파라미터를 인식하지 못한다는 오류가 발생하면 먼저 명령의 형식이 올바른지 확인합니다. 형식이 올바른 경우 AWS CLI의 최신 버전으로 업그레이드하는 것이 좋습니다. 의 업데이트된 버전 AWS CLI 은 거의 모든 영업일에 릴리스됩니다. 새로운 AWS 서비스, 기능 및 파라미터가 이러한 새 버전의 에 도입되었습니다 AWS CLI. 새로운 서비스, 기능 또는 파라미터에 액세스할 수 있는 유일한 방법은 해당 요소가 도입된 이후 릴리스된 버전으로 업그레이드하는 것입니다.

의 버전을 업데이트하는 방법은 에 설명된 대로 를 처음 설치한 방법에 AWS CLI 따라 달라집니다설치 AWS CLI.

번들 설치 관리자 중 하나를 사용한 경우 운영 체제에 적합한 최신 버전을 다운로드하여 설치하기 전에 기존 설치를 제거해야 할 수 있습니다.

맨 위로 이동

## --debug 옵션 사용

가 즉시 이해하지 못하는 오류를 AWS CLI 보고하거나 예상치 못한 결과를 생성하면 --debug 옵션을 사용하여 명령을 다시 실행하여 오류에 대한 자세한 내용을 확인할 수 있습니다. 이 옵션을 사용하면 AWS CLI 가 명령을 처리하는 데 필요한 모든 단계에 대한 세부 정보를 출력합니다. 출력에 있는 세부 정보를 통해 오류가 언제 발생했고 어디서 시작되었는지에 대한 단서를 확인할 수 있습니다.

이후 검토를 위해 출력을 텍스트 파일로 보내거나 요청이 있을 때 출력을 AWS Support 에 보낼 수 있습니다.

--debug 옵션을 포함하면 다음과 같은 세부 정보가 포함됩니다.

- 보안 인증 검색
- 제공된 파라미터 구문 분석
- AWS 서버로 전송된 요청 구성
- 로 전송된 요청의 내용 AWS
- 원시 응답의 내용
- 형식이 지정된 출력

다음은 --debug 옵션을 사용할 때와 사용하지 않을 때의 명령 실행의 예입니다.

```
$ 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/
paginators-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
5f776d91EXAMPLE9b8cb5eb5d6d4a787a33ae41c8cd6eEXAMPLEca69080e1e1f
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:
d85a0EXAMPLEb40164f2f539cdc76d4f294fe822EXAMPLE18ad1ddf58a1a3ce7
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'<ListGroupsResponse xmlns="https://iam.amazonaws.com/doc/2010-05-08/">\n
 <ListGroupsResult>\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  </ListGroupsResult>\n
 <ResponseMetadata>\n    <RequestId>74c11606-bd38-11e9-9c82-559da0adb349</RequestId>\n
 </ResponseMetadata>\n</ListGroupsResponse>\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"
```

```
        }
    ]
}
```

맨 위로 이동

## AWS CLI 명령 기록 로그 활성화 및 검토

`cli_history` 파일 설정을 사용하여 AWS CLI 명령 기록 로그를 활성화할 수 있습니다. 이 설정을 활성화하면 는 `aws` 명령 기록을 AWS CLI 기록합니다.

이 기록을 나열하려면 `aws history list` 명령을 사용하고 세부 정보를 보려면 `aws history show` 명령에 결과 command_ids를 사용할 수 있습니다. 자세한 내용은 AWS CLI 참조 가이드의 `aws history` 섹션을 참조하세요.

`--debug` 옵션을 포함하면 다음과 같은 세부 정보가 포함됩니다.

- API botocore 호출
- 상태 코드
- HTTP 응답
- 헤더
- 반환 코드

이 정보를 사용하여 파라미터 데이터와 API 호출이 예상대로 작동하는지 확인한 다음 명령이 실패하는 프로세스의 단계를 추론할 수 있습니다.

맨 위로 이동

## AWS CLI 가 구성되었는지 확인

`config` 및 `credentials` 파일 또는 IAM 사용자 또는 역할이 올바르게 구성되지 않은 경우 다양한 오류가 발생할 수 있습니다. `config` 및 `credentials` 파일 또는 IAM 사용자 또는 역할의 오류 해결에 대한 자세한 내용은 the section called “액세스 거부 오류” 및 섹션을 참조하세요the section called “잘못된 보안 인증 정보 및 키 오류”.

맨 위로 이동

# 명령을 찾을 수 없음 오류

이 오류는 운영 체제가 AWS CLI 명령을 찾을 수 없음을 의미합니다. 설치가 불완전하거나 업데이트가 필요할 수 있습니다.

가능한 원인: 설치된 버전보다 최신 AWS CLI 기능을 사용하려고 하거나 형식이 잘못되었습니다.

오류 텍스트 예:

```
$ 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
....
```

명령의 형식이 잘못되었거나 기능이 릴리스되기 전의 이전 버전을 사용하는 경우 여러 가지 오류가 발생할 수 있습니다. 이 두 가지 문제를 해결하는 방법에 대한 자세한 내용은 the section called “AWS CLI 명령 형식 확인” 및 the section called “최신 버전의 AWS CLI를 실행 중인지 확인합니다.” 섹션을 참조하세요.

맨 위로 이동

가능한 원인: 설치 후 터미널을 다시 시작해야 함

오류 텍스트 예:

```
$ aws --version
command not found: aws
```

를 처음 설치하거나 업데이트한 후 aws 명령을 찾을 수 없는 경우 AWS CLI업데이트를 인식하려면 터미널을 다시 시작해야 할 수 있습니다PATH.

맨 위로 이동

가능한 원인: 이 완전히 설치되지 AWS CLI 않음

오류 텍스트 예:

```
$ aws --version
command not found: aws
```

를 처음 설치하거나 업데이트한 후 aws 명령을 찾을 수 없는 경우 완전히 설치되지 않았을 AWS CLI수 있습니다. 플랫폼에 해당하는 설치 AWS CLI 단계에 따라 다시 설치해 봅니다.

맨 위로 이동

가능한 원인: AWS CLI 에 권한이 없음(Linux)

AWS CLI Linux에 를 처음 설치하거나 업데이트한 후 aws 명령을 찾을 수 없는 경우 설치된 폴더에 대한 execute 권한이 없을 수 있습니다. AWS CLI 설치에 를 사용하여 다음 명령을 실행PATH하여 에 chmod 권한을 부여합니다 AWS CLI.

```
$ sudo chmod -R 755 /usr/local/aws-cli/
```

맨 위로 이동

가능한 원인: 설치하는 동안 운영 체제 **PATH**가 업데이트되지 않음

오류 텍스트 예:

```
$ aws --version
command not found: aws
```

aws 실행 파일을 운영 체제의 PATH 환경 변수에 추가해야 할 수 있습니다. 를 AWS CLI 에 추가하려면 운영 체제에 대해 다음 지침을 PATH사용합니다.

Linux and macOS

1. 사용자 디렉터리에서 셸의 프로파일 스크립트를 찾습니다. 어떤 셸을 가지고 있는지 잘 모르는 경우 echo $SHELL을 실행합니다.

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

   - Bash - .bash_profile, .profile 또는 .bash_login
   - Zsh – .zshrc

- Tcsh - `.tcshrc`, `.cshrc` 또는 `.login`

2. 내보내기 명령을 프로필 스크립트에 추가하세요. 다음 명령은 현재 PATH 변수에 로컬 bin을 추가합니다.

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

3. 현재 세션에 업데이트된 프로파일을 다시 로드합니다.

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

Windows

1. Windows 명령 프롬프트에서 `where` 명령을 `/R` *path* 파라미터와 함께 사용하여 `aws` 파일 위치를 찾습니다. `aws`를 포함한 모든 폴더가 반환됩니다.

```
C:\> where /R c:\ aws
c:\Program Files\Amazon\AWSCLIV2\aws.exe
...
```

기본적으로 AWS CLI 버전 2는 다음 위치에 있습니다.

```
c:\Program Files\Amazon\AWSCLIV2\aws.exe
```

2. Windows 키를 누르고 **environment variables**를 입력하세요.
3. 제안 목록에서 Edit environment variables for your account를 선택합니다.
4. PATH를 선택한 다음 편집을 선택합니다.
5. 첫 번째 단계에서 찾은 경로(예: *C:\Program Files\Amazon\AWSCLIV2\aws.exe*)를 Variable value 필드에 추가합니다.
6. 확인을 두 번 선택하여 새 설정을 적용합니다.
7. 실행 중인 명령 프롬프트를 모두 닫았다가 명령 프롬프트 창을 다시 엽니다.

맨 위로 이동

# 'aws --version' 명령이 설치한 버전과 다른 버전을 반환함

터미널이 예상 AWS CLI 과 다른 에 PATH 대해 를 반환할 수 있습니다.

가능한 원인: 설치 후 터미널을 다시 시작해야 함

aws 명령에 잘못된 버전이 표시되는 경우 PATH 업데이트를 인식하도록 터미널을 다시 시작해야 할 수 있습니다. 활성 터미널뿐만 아니라 열려 있는 모든 터미널을 닫아야 합니다.

맨 위로 이동

가능한 원인: 설치 후 시스템을 다시 시작해야 함

aws 명령에 잘못된 버전이 표시되고 터미널을 다시 시작해도 문제가 해결되지 않는 경우 PATH 업데이트를 인식하도록 시스템을 다시 시작해야 할 수 있습니다.

맨 위로 이동

가능한 원인: 여러 버전의 가 있습니다. AWS CLI

를 업데이트 AWS CLI 하고 기존 설치와 다른 설치 방법을 사용한 경우 여러 버전이 설치될 수 있습니다. 예를 들어 Linux 또는 macOS에서 현재 설치에 pip를 사용했지만 .pkg 설치 파일을 사용하여 업데이트를 시도한 경우 특히 이전 버전을 가리키는 PATH와 충돌이 발생할 수 있습니다.

이 문제를 해결하려면 모든 버전의 AWS CLI를 제거하고 새로 설치를 수행합니다.

모든 버전을 제거한 후 운영 체제에 해당하는 지침을 따라 AWS CLI 버전 1 또는 AWS CLI 버전 2의 원하는 버전을 설치합니다.

> **Note**
>
> AWS CLI 버전 1이 이미 설치되어 버전 AWS CLI 2를 설치한 후 이 문제가 발생하면 AWS CLI 버전 1에서 마이그레이션할 때 설치 지침 의 마이그레이션 지침을 따릅니다.

맨 위로 이동

## “aws --version” 명령은 를 제거한 후 버전을 반환합니다. AWS CLI

이는 시스템에 가 아직 AWS CLI 설치되어 있을 때 발생하는 경우가 많습니다.

가능한 원인: 제거 후 터미널을 다시 시작해야 함

aws --version 명령이 여전히 작동하는 경우 터미널 업데이트를 인식하도록 터미널을 다시 시작해야 할 수 있습니다.

맨 위로 이동

가능한 원인: AWS CLI 시스템에 여러 버전의 가 있거나 원래 설치에 사용한 것과 동일한 제거 방법을 사용하지 않았습니다. AWS CLI

설치에 사용한 것과 다른 방법을 AWS CLI 사용하여 를 제거하거나 여러 버전을 설치한 경우 가 올바르게 제거되지 AWS CLI 않을 수 있습니다. 예를 들어 현재 설치에 `pip`를 사용한 경우 `pip`를 사용하여 제거해야 합니다. 이 문제를 해결하려면 설치에 사용한 것과 동일한 방법을 AWS CLI 사용하여 를 제거합니다.

1. 운영 체제 및 원래 설치 방법에 해당하는 지침을 따라 AWS CLI 버전 1 및 AWS CLI 버전 2를 제거합니다.
2. 열려 있는 터미널을 모두 닫습니다.
3. 원하는 터미널을 열고 다음 명령에 입력한 후 버전이 반환되지 않는지 확인합니다.

```
$ aws --version
command not found: aws
```

출력에 여전히 버전이 나열된 경우 가 다른 방법을 사용하여 설치 AWS CLI 되었거나 버전이 여러 개 있을 가능성이 높습니다. 를 설치한 메서드를 모르는 경우 AWS CLI버전 출력이 수신되지 않을 때까지 운영 체제에 적합한 AWS CLI 버전 1 및 AWS CLI 버전 2의 각 제거 메서드에 대한 지침을 따릅니다.

> **Note**
>
> 패키지 관리자를 사용하여 AWS CLI (`pip`, `apt`, `brew` 등)를 설치한 경우 동일한 패키지 관리자를 사용하여 제거해야 합니다. 모든 버전의 패키지를 제거하는 방법에 대해 패키지 관리자가 제공하는 지침을 따르세요.

맨 위로 이동

# 이 불완전한 파라미터 이름을 가진 명령을 AWS CLI 처리했습니다.

가능한 원인: AWS CLI 파라미터의 알려진 약어를 사용했습니다.

AWS CLI 는 Python을 사용하여 구축되므로 는 allow_abbrev 인수를 포함하여 Python argparse 라이브러리를 AWS CLI 사용합니다. 파라미터의 약어는 에서 인식하고 AWS CLI 처리합니다.

다음 create-change-set 명령 예제는 CloudFormation 스택 이름을 변경합니다. 파라미터는 의 약어로 인식되며 --change-set-name--change-set-n는 명령을 AWS CLI 처리합니다.

```
$ aws cloudformation create-change-set --stack-name my-stack --change-set-n my-change-set
```

약어가 여러 명령일 수 있는 경우 파라미터는 약어로 인식되지 않습니다.

다음 create-change-set 명령 예제는 CloudFormation 스택 이름을 변경합니다. --change-set-name 및 --change-set-type와 같이 약어가 될 수 있는 여러 파라미터가 있기 때문에 --change-set- 파라미터는 약어로 인식되지 않습니다. 따라서 AWS CLI 는 명령을 처리하지 않습니다.

```
$ aws cloudformation create-change-set --stack-name my-stack --change-set- my-change-set
```

**Warning**

의도적으로 파라미터 약어를 사용하지 마세요. 신뢰할 수 없으며 이전 버전과도 호환되지 않습니다. 약어를 혼동시키는 새 파라미터가 명령에 추가되면 명령이 손상됩니다.
또한 파라미터가 단일 값 인수인 경우 명령에서 예상치 못한 동작이 발생할 수 있습니다. 단일 값 인수의 여러 인스턴스가 전달되면 마지막 인스턴스만 실행됩니다. 다음 예에서 --filters 파라미터는 단일 값 인수를 사용합니다. --filters 및 --filter 파라미터는 지정됩니다. --filter 파라미터는 --filters의 약어입니다. 이로 인해 --filters의 두 인스턴스가 적용되고 마지막 --filter 인수만 적용됩니다.

```
$ aws ec2 describe-vpc-peering-connections \
    --filters Name=tag:TagName,Values=VpcPeeringConnection \
    --filter Name=status-code,Values=active
```

명령을 실행하기 전에 올바른 파라미터를 사용하고 있는지 확인하여 예기치 않은 동작을 방지하세요.

맨 위로 이동

## 액세스 거부 오류

가능한 원인: AWS CLI 프로그램 파일에 '실행' 권한이 없습니다.

Linux 또는 macOS에서 aws 프로그램이 호출하는 사용자에 대한 실행 권한을 가지고 있는지 확인합니다. 일반적으로 사용 권한은 755로 설정됩니다.

사용자의 실행 권한을 추가하려면 다음 명령을 실행하여 *~/.local/bin/aws* 컴퓨터의 프로그램 경로가 있습니다.

```
$ chmod +x ~/.local/bin/aws
```

맨 위로 이동

가능한 원인: 자격 IAM 증명에 작업을 수행할 권한이 없습니다.

오류 텍스트 예:

```
$ aws s3 ls
An error occurred (AcessDenied) when calling the ListBuckets operation: Access
 denied.
```

AWS CLI 명령을 실행하면 IAM 사용자를 계정 또는 역할과 연결하는 보안 인증 정보를 사용하여 사용자를 대신하여 AWS 작업이 수행됩니다. 연결된 정책은 에서 실행하는 명령에 해당하는 API 작업을 호출할 수 있는 권한을 부여해야 합니다 AWS CLI.

대부분의 명령은 명령 이름과 일치하는 이름으로 한 가지 작업을 호출합니다. 그러나 와 같은 사용자 지정 명령은 여러 을 `aws s3 sync` 호출합니다APIs. `--debug` 옵션을 사용하여 명령APIs이 호출하는 항목을 확인할 수 있습니다.

사용자 또는 역할에 정책에서 할당한 적절한 권한이 있다고 확신하는 경우 AWS CLI 명령이 예상한 자격 증명을 사용하고 있는지 확인합니다. 자격 증명에 대한 다음 섹션을 참조하여 에서 사용 AWS CLI 중인 자격 증명이 예상한 자격 증명인지 확인합니다.

IAM 권한 할당에 대한 자세한 내용은 IAM 사용 설명서의 액세스 관리 개요: 권한 및 정책을 참조하세요.

맨 위로 이동

## 잘못된 보안 인증 정보 및 키 오류

오류 텍스트 예:

```
$ 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.
```

가능한 원인: 이 잘못된 보안 인증 정보를 읽거나 예기치 않은 위치에서 읽 AWS CLI 습니다.

가 예상과 다른 위치에서 자격 증명을 읽거나 키 페어 정보가 올바르지 않을 AWS CLI 수 있습니다. `aws configure list`를 실행하여 어떤 보안 인증을 사용하고 있는지 확인할 수 있습니다.

다음은 기본 프로파일에 사용된 보안 인증을 확인하는 방법을 나타낸 예제입니다.

```
$ 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
```

다음은 명명된 프로파일의 보안 인증을 확인하는 방법을 나타낸 예제입니다.

```
$ 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
```

키 페어 세부 정보를 확인하려면 `config` 및 `credentials` 파일을 검토합니다. `config` 및 `credentials` 파일에 대한 자세한 내용은 the section called “의 구성 및 보안 인증 파일 설정 AWS CLI” 섹션을 참조하세요. 보안 인증 정보 우선 순위를 비롯한 인증 및 보안 인증에 대한 자세한 내용은 인증 및 액세스 보안 인증 섹션을 참조하세요.

맨 위로 이동

가능한 원인: 컴퓨터의 클록이 동기화되지 않음

유효한 보안 인증 정보를 사용 중이라면 클록이 동기화되지 않았을 수 있습니다. Linux 또는 macOS에서 `date`를 실행하여 시간을 확인합니다.

```
$ date
```

몇 분 안에 시스템 클록이 정확하지 않으면 `ntpd`를 사용하여 동기화합니다.

```
$ sudo service ntpd stop
$ sudo ntpdate time.nist.gov
$ sudo service ntpd start
$ ntpstat
```

Windows에서는 제어판의 날짜 및 시간 옵션을 사용하여 시스템 클록을 구성합니다.

맨 위로 이동

## 서명 불일치 오류

오류 텍스트 예:

```
$ 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.
```

이 명령을 AWS CLI 실행하면 적절한 AWS 서비스 작업을 수행하기 위해 AWS 서버에 암호화된 요청을 보냅니다. 보안 인증 정보(액세스 키 및 보안 암호 키)는 암호화에 관여하며 가 요청을 수행하는 사람을 인증 AWS 할 수 있도록 합니다. 다음과 같이 이 프로세스의 올바른 작업에 방해가 될 수 있는 요소가 여러 개 있습니다.

가능한 원인: 시계가 AWS 서버와 동기화되지 않음

재생 공격(Replay Attack)으로부터 보호하기 위해 암호화/암호 해독 프로세스 동안 현재 시간이 사용될 수 있습니다. 클라이언트 및 서버의 시간이 허용된 시간을 넘는 경우 프로세스가 실패할 수 있으며 요청이 거부됩니다. 이는 클록이 호스트 머신의 클록과 동기화되지 않은 가상 머신에서 명령을 실행할 때에도 발생할 수 있습니다. 한 가지 가능한 원인은 가상 머신이 최대 절전 모드에 있다가 활성화된 후 얼마 뒤 클록을 호스트 머신과 동기화할 때입니다.

Linux 또는 macOS에서 `date`를 실행하여 시간을 확인합니다.

```
$ date
```

몇 분 안에 시스템 클록이 정확하지 않으면 `ntpd`를 사용하여 동기화합니다.

```
$ sudo service ntpd stop
$ sudo ntpdate time.nist.gov
$ sudo service ntpd start
$ ntpstat
```

Windows에서는 제어판의 날짜 및 시간 옵션을 사용하여 시스템 클록을 구성합니다.

맨 위로 이동

가능한 원인: 운영 체제에서 특정 특수 문자가 포함된 AWS 키를 잘못 처리하고 있습니다.

AWS 키에 , -, + /또는 와 같은 특정 특수 문자가 포함된 경우 %일부 운영 체제 변형은 문자열을 부적절하게 처리하고 키 문자열을 잘못 해석합니다.

생성 과정에서 새 인스턴스에 보안 인증 파일을 빌드하는 도구와 같은 다른 도구 또는 스크립트를 사용하여 키를 처리하는 경우 이러한 도구 및 스크립트는 특수 문자를 자체적으로 처리하여 더 이상 인식 AWS 하지 못하는 것으로 변환될 수 있습니다.

문제를 일으키는 특수 문자가 포함되지 않은 비밀 키를 얻으려면 비밀 키를 다시 생성하는 것이 좋습니다.

맨 위로 이동

# Windows 콘솔을 찾을 수 없음 오류

오류 텍스트 예:

```
$ aws s3 ls
No Windows console found. Are you running cmd.exe?
```

AWS CLI 명령을 사용하면 “Windows 콘솔을 찾을 수 없습니다. cmd.exe를 실행 중입니까?” 오류 메시지를 받게 됩니다. 이는 일반적으로 설치prompt_toolkit한 Python이 오래된 경우 AWS CLI 버전 1에 표시되는 오류입니다. 이 문제를 해결하려면 Python 웹 사이트에서 prompt_toolkit 최신 버전을 설치하세요.

맨 위로 이동

# SSL 인증서 오류

가능한 원인: AWS CLI 가 프록시의 인증서를 신뢰하지 않음

오류 텍스트 예:

```
$ aws s3 ls
[SSL: CERTIFICATE_ VERIFY_FAILED] certificate verify failed
```

AWS CLI 명령을 사용하면 [SSL: CERTIFICATE_ VERIFY_FAILED] certificate verify failed 오류 메시지가 표시됩니다. 이는 프록시 인증서가 자체 서명되고 회사가 인증 기관(CA)으로 설정된 등의 요인으로 인해 프록시 인증서를 신뢰 AWS CLI 하지 못하기 때문입니다. 이렇게 하면 AWS CLI 가 로컬 CA 레지스트리에서 회사 CA 루트 인증서를 찾을 수 없습니다.

이 문제를 해결하려면 ca_bundle 구성 .pem 파일 설정, --ca-bundle 명령줄 옵션 또는 AWS_CA_BUNDLE 환경 변수를 사용하여 회사 파일을 찾을 수 AWS CLI 있는 위치를 에 지시합니다.

맨 위로 이동

가능한 원인: 구성이 올바른 CA 루트 인증서 위치를 가리키지 않음

오류 텍스트 예:

```
$ aws s3 ls
```

```
SSL validation failed for regionname [Errno 2] No such file or directory
```

이것은 AWS CLI에서 인증 기관(CA) 번들 파일 위치가 잘못 구성되었기 때문에 발생합니다. 이 문제를 해결하려면 회사 `.pem` 파일의 위치를 확인하고 `ca_bundle` 구성 파일 설정, `--ca-bundle` 명령줄 옵션 또는 `AWS_CA_BUNDLE` 환경 변수를 사용하여 AWS CLI 구성을 업데이트합니다.

맨 위로 이동

가능한 원인: 구성이 올바른 를 사용하지 않습니다. AWS 리전

오류 텍스트 예:

```
$ aws s3 ls
[SSL: CERTIFICATE_ VERIFY_FAILED] certificate verify failed
```

지정된 에 를 사용할 수 AWS 서비스 없거나 리소스가 다른 에 있는 경우 오류 AWS 리전 또는 예상치 못한 결과가 발생할 수 있습니다 AWS 리전. 문제 해결 단계는 the section called “AWS CLI 명령이 사용 중인 AWS 리전 지 확인” 섹션을 참조하세요.

맨 위로 이동

가능한 원인: TLS 버전을 업데이트해야 함

오류 텍스트 예:

```
$ aws s3 ls
[SSL: UNSAFE_LEGACY_RENEGOTIATION_DISABLED] unsafe legacy renegotiation disabled
```

AWS 서비스 는 디바이스 버전과 호환되지 TLS 않는 TLS 버전을 사용하고 있습니다. 이 문제를 해결하려면 지원되는 TLS 버전으로 업데이트하세요. 자세한 내용은 the section called “최소 TLS 버전 적용” 단원을 참조하십시오.

맨 위로 이동

## 잘못된 JSON 오류

오류 텍스트 예:

```
$ 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}
```

AWS CLI 명령을 사용하면 "Invalid JSON" 오류 메시지가 표시됩니다. 이는 일반적으로 예상 JSON 형식의 명령을 입력할 때 나타나는 오류이며 가 를 JSON 올바르게 읽을 AWS CLI 수 없습니다.

가능한 원인: 사용할 에 JSON 대해 유효한 AWS CLI 를 입력하지 않았습니다.

명령에 유효한 를 JSON 입력했는지 확인합니다. 형식 지정에 문제가 있는 JSON 경우 JSON 검사기를 사용하는 것이 좋습니다.

명령줄에서 더 고급JSON으로 사용하려면 와 같은 명령줄 JSON 프로세서를 사용하여 JSON 문자열을 생성하는 jq것이 좋습니다. 에 대한 자세한 내용은 의 jq 리포지토리를 jq참조하세요GitHub.

맨 위로 이동

가능한 원인: 터미널의 견적 규칙으로 인해 로 유효한 전송이 불가능할 JSON 수 있습니다. AWS CLI

이 명령에서 무언가를 AWS CLI 수신하기 전에 터미널은 자체 인용 및 이스케이프 규칙을 사용하여 명령을 처리합니다. 터미널의 형식 지정 규칙으로 인해 명령이 에 전달되기 전에 일부 JSON 콘텐츠가 제거될 수 있습니다 AWS CLI. 명령을 공식화할 때 터미널의 인용 규칙을 사용해야 합니다.

문제를 해결하려면 echo 명령을 사용하여 셸에서 파라미터를 처리하는 방법을 확인합니다.

```
$ echo {"ReadCapacityUnits":15,"WriteCapacityUnits":10}
ReadCapacityUnits:15 WriteCapacityUnits:10
```

```
$ echo '{"ReadCapacityUnits":15,"WriteCapacityUnits":10}'
{"ReadCapacityUnits":15,"WriteCapacityUnits":10}
```

유효한 JSON 이 반환될 때까지 명령을 수정합니다.

보다 심층적인 문제 해결을 위해 --debug 파라미터를 사용하여 디버그 로그를 확인합니다. 디버그 로그에는 AWS CLI에 전달된 내용이 정확히 표시되어 있습니다.

```
$ 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']
```

터미널의 견적 규칙을 사용하여 JSON 입력이 로 전송될 때 발생하는 모든 문제를 해결합니다 AWS CLI. 인용 규칙에 대한 자세한 내용은 the section called “문자열과 따옴표” 섹션을 참조하세요.

> **Note**
>
> 에 대한 유효성을 얻는 JSON 데 문제가 있는 경우 Blobs를 사용하여 JSON 데이터를 에 직접 전달하여 JSON 데이터 입력에 대한 터미널의 견적 규칙을 우회하는 것이 AWS CLI좋습니다 AWS CLI. Blob에 대한 자세한 내용은 Blob 섹션을 참조하세요.

맨 위로 이동

## 추가 리소스

AWS CLI 문제에 대한 추가 도움이 필요하면 GitHub 또는 AWS CLI 커뮤니티 AWS re:Post 에서 커뮤니티를 참조하세요.

맨 위로 이동

# AWS CLI 사용 설명서 문서 기록

다음 표에는 2019년 1월 이후 AWS Command Line Interface 사용 설명서에 대한 중요 추가 사항이 설명되어 있습니다. 이 설명서에 대한 업데이트 알림을 받으려면 RSS 피드를 구독하면 됩니다.

| 변경 사항 | 설명 | 날짜 |
| --- | --- | --- |
| 보안 인증 및 인증 정보가 업데이트되었습니다. | 보안 인증 및 인증 방법 지침 및 예제가 업데이트되었습니다. 여기에는 관련 구성 페이지 업데이트가 포함됩니다. 설명서가 늘어남에 따라 관련 보안 인증 항목이 새로운 인증 및 액세스 보안 인증 섹션으로 이동되었습니다. | 2023년 3월 31일 |
| AWS CLI V1과 V2에 대한 콘텐츠가 이제 각각의 가이드로 분리됨 | 명확성과 사용 편의성을 위해 AWS CLI 버전 1과 AWS CLI 버전 2 콘텐츠가 이제 자체적인 가이드로 분리되어 있습니다. 를 참조하세요. AWS CLI 버전 2의 경우 최신 AWS Command Line Interface 사용 설명서를 참조하세요. | 2021년 11월 2일 |
| AWS CLI 별칭 정보가 추가됨 | AWS CLI 별칭 정보를 추가했습니다. 별칭은 자주 사용하는 명령이나 스크립트를 단축하기 위해 AWS Command Line Interface(AWS CLI)에서 생성할 수 있는 바로 가기입니다. | 2021년 3월 11일 |
| 필터 출력 정보 업데이트됨 | 필터에 대한 정보를 업데이트하고 해당 페이지로 이동했습니다. | 2021년 2월 1일 |

| | | |
|---|---|---|
| Python 2.7, 3.4, 3.5의 사용 중지 발표 | Python 2.7은 2020년 1월 1일 Python Software Foundation에 의해 사용 중지되었습니다. 앞으로 AWS CLI 버전 1을 사용하는 고객은 Python 3(최소 Python 3.6)을 사용하도록 전환해야 합니다. Python 2.7 지원은 2021년 7월 19일부터 AWS CLI 버전 1의 새 버전에 대해 사용 중지됩니다. Python 3.4 및 3.5는 2021년 2월 1일부터 더 이상 사용되지 않습니다. | 2021년 1월 29일 |
| Amazon S3 스크립팅 예제 추가됨 | Amazon S3 수명 주기 스크립팅 예제를 추가했습니다. | 2020년 10월 15일 |
| Amazon EC2 스크립팅 예제 추가됨 | Amazon EC2 인스턴스 유형 스크립팅 예제를 추가했습니다. | 2020년 10월 15일 |
| 재시도 정보 추가됨 | AWS CLI 기능 및 동작에 대한 재시도 페이지를 추가했습니다. | 2020년 9월 17일 |
| 서버 측 및 클라이언트 측 페이지 매김 페이지 | 페이지 매김 정보를 업데이트하고 한 페이지에 중앙화했습니다. | 2020년 8월 17일 |
| s3 명령 페이지 업데이트됨 | 새 예제 및 리소스로 상위 수준 s3 명령 페이지를 업데이트했습니다. | 2020년 7월 30일 |
| 업데이트된 설치 정보 | Linux, macOS 및 Windows의 설치, 업데이트 및 제거 정보가 업데이트됩니다. | 2020년 5월 19일 |

| | | |
|---|---|---|
| AWS CLI 버전 1에서 Python 2.6 및 3.3에 대한 지원이 제거되어 문서가 업데이트됨 | 2020년 1월 10일 현재 AWS CLI 버전 1은 더 이상 Python 버전 2.6 또는 3.3 사용을 지원하지 않습니다. AWS CLI 버전 1.17 이상을 사용하려면 최신 버전의 Python으로 업데이트해야 합니다. | 2020년 1월 10일 |
| 새 MFA 단원 | 멀티 팩터 인증 및 역할을 사용하여 CLI에 액세스하는 방법을 설명하는 새 단원이 추가되었습니다. | 2019년 5월 3일 |
| "CLI 사용" 단원 업데이트 | CLI 사용 지침 및 절차에 대한 주요 개선 사항 및 추가 사항입니다. | 2019년 3월 7일 |
| "CLI 설치" 단원 업데이트 | AWS CLI 설치 지침 및 절차에 대한 주요 개선 사항 및 추가 사항입니다. | 2019년 3월 7일 |
| "CLI 구성" 단원 업데이트 | AWS CLI 구성 지침 및 절차에 대한 주요 개선 사항 및 추가 사항입니다. | 2019년 3월 7일 |